Un livre de Wikilivres.

Microsoft SQL Server

Introduction

Présentation

Microsoft SQL Server (alias MSSQL) est un système de gestion de base de données (SGBD) développé par la société Microsoft.

Son extension du langage SQL est appelée Transact-SQL (T-SQL).

Installation du serveur

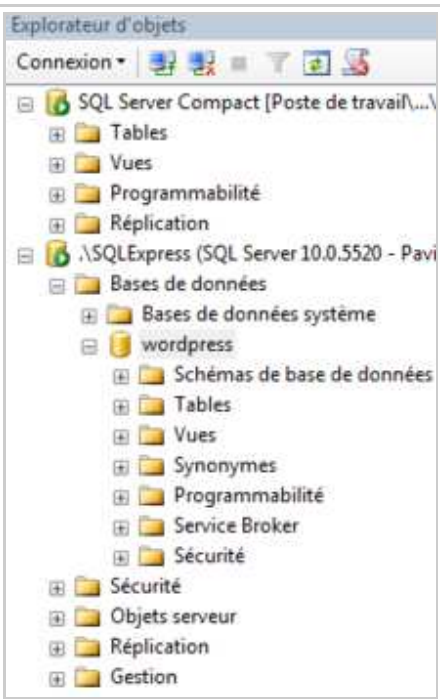
Ce logiciel n'est disponible que sur le système d'exploitation Microsoft Windows.

Connexion de SQL Server Management Studio à une base SQLExpress.

1. La version gratuite s'appelle SQL Server Express Télécharger (https://msdn.microsoft.com/fr-fr/sqlserver2014express.aspx) [archive]

   (anciennement Microsoft SQL Server Desktop Engine (MSDE). Elle permet de créer et manipuler des bases de 2 Go maximum, et de se connecter à d'autres serveurs de base de données existants[1].
2. La version payante nécessite d'acheter soit une licence pour le serveur (à partir de 900 €) plus une par ordinateur client (autour de 15 €), soit une licence par processeur à partir de 4 000 €[2]. Elle permet d'utiliser des bases jusqu'à 16 To avec des tables à 30 000 colonnes[3].

   Remarque : il existe aussi une édition compacte encore plus limitée (ex : aucune option de de sécurité).



Menu avec une base SQL Compact puis une SQLExpress.

Pour PC

> **Attention !**
>
> SQL Server se lance ensuite automatiquement à chaque démarrage de la machine, ce qui la ralentit significativement.

Pour éviter cela, exécuter *services.msc*, puis passer le service *MSSQL$SQLEXPRESS* en démarrage manuel. Ensuite pour lancer le service à souhait (en tant qu'administrateur), créer un script *SQL_Server.cmd* contenant la ligne suivante :

```
net start "MSSQL$SQLEXPRESS"
```

PHP

Pour se connecter au serveur MS-SQL à partir d'un tout-en-un comme EasyPHP, il convient de ne pas prendre la version la plus récente de PHP mais plutôt :

- Sous PHP 4, il suffisait de copier le fichier `php_mssql.dll` dans les extensions.
- Pour PHP 5.4 :
  1. Télécharger les .dll sur le site officiel[4] (SQL30)
  2. Les copier dans `C:\PROGRA~2\EasyPHP\binaries\php\php_runningversion\ext`
  3. Les ajouter dans `C:\PROGRA~2\EasyPHP\binaries\php\php_runningversion\php.ini` via les lignes suivantes[5] :
     ```
     extension=php_sqlsrv_54_ts.dll
     extension=php_pdo_sqlsrv_54_ts.dll
     ```
  4. Redémarrer EasyPHP puis vérifier que la ligne pdo_sqlsrv s'est bien ajoutée dans `http://127.0.0.1/home/index.php?page=php-page&display=extensions`
- Dans PHP 5.5 on obtient toujours `Fatal error: Call to undefined function sqlsrv_connect()`.


Logo EasyPHP.

Sinon, voir Internet Information Services.

ODBC

En lançant %windir%\system32\odbcad32.exe il est possible de configurer une liaison vers le serveur MS-SQL dans l'onglet "Source de données système".

L'interface SQL Server Management Studio

La version 2008 de Server Management Studio (SSMS) peut se télécharger en même temps que le logiciel, ou individuellement sur https://www.microsoft.com/fr-fr/download/confirmation.aspx?id=7593.

Lancer depuis le menu démarrer le programme *SQL Server Management Studio*.

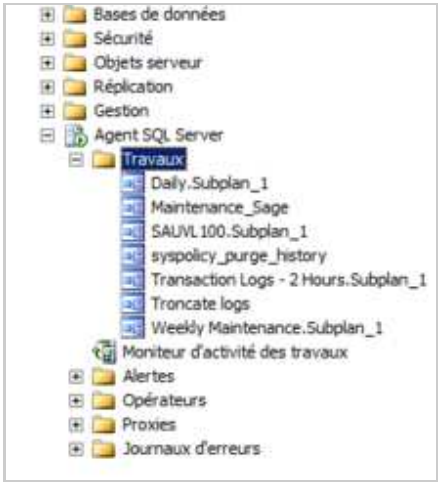
Le logiciel a ensuite besoin de se connecter au serveur de base de données (ex : localhost), avec mot de passe.

Navigation

Le logiciel permet de créer et faire dérouler tous les éléments de chaque base grâce à son explorateur d'objet sur la gauche :

- *Base1*
  - *Tables*
  - *Procédures stockées*
- *Base2*

...



Menu travaux contenant plusieurs jobs.

Attention !

> Copier une base de données remplace le propriétaire de l'originale par *AUTORITE NT / SYSTEM*. Il faut ensuite lancer une commande `ALTER AUTHORIZATION`[6] pour rétablir l'initial.

Ses fonctions de recherche sont limitées à l'option "filtrer" (icône d'entonnoir). Pour recherche une table dans plusieurs bases il faut donc entrer une requête SQL (voir chapitre suivante).

Le menu "Travaux" situé sous les bases permet de programmer des tâches planifiées (ex : troncature de logs ou de tables).

Selon l'emplacement courant, certaines barres d'outils s'affichent ou se masquent automatiquement.

Schéma

Chaque base de données peut contenir plusieurs schémas : les diagrammes de classes. En effet, SSMS permet d'y afficher les tables existantes avec leurs champs, et d'y ajouter des index et des relations.


Interface d'ajout de relation

Pour créer une liaison entre deux tables, faire un clic droit sur l'une, puis "Relations". Dans la fenêtre apparue, cliquer sur "Ajouter". Une relations temporaire apparait alors et il convient de la modifier[7] :

- Cliquer sur les points de suspension de la ligne "Spécification de tables et colonnes", pour sélectionner les clés à relier.
- Si le lien est simplement créé pour les besoins du dessins, passer le champ "Appliquer la contrainte de clé étrangère" à "Non".

Références

1. http://technet.microsoft.com/fr-fr/library/bb967613.aspx
2. http://blogs.developpeur.org/christian/archive/2011/09/12/Prix-sql-server-en-france-pour-SQL-Server-2008-R2.aspx
3. http://technet.microsoft.com/fr-fr/library/ms143432.aspx
4. http://www.microsoft.com/en-us/download/details.aspx?id=20098
5. http://www.php.net/manual/fr/ref.pdo-sqlsrv.php
6. https://msdn.microsoft.com/fr-fr/library/ms187359%28v=SQL.120%29.aspx
7. https://msdn.microsoft.com/fr-fr/library/ms189049%28v=sql.120%29.aspx?f=255&MSPPError=-2147217396

Voir aussi

- *(anglais)* Vidéo officielle d'installation de SQL 2008 (http://msdn.microsoft.com/fr-fr/library/dd299415%28v=sql.100%29.aspx) [archive] (sous-titrée)
- *(anglais)* Installing WordPress on SQL Server (http://wordpress.visitmix.com/development/installing-wordpress-on-sql-server#prerequisites) [archive]

Bases de données

Bases de données système

Après installation, le serveur fournit quatre bases de données système, indispensable à son bon fonctionnement :

1. *master* : enregistre les informations du serveur SQL dans des vues du système.
2. *model* : modèle pour créer des bases vierge.
3. *msdb* : stocke les tâches planifiées.
4. *tempdb* : ressources temporaires, utilisables pas tous les utilisateurs.

Elles peuvent donc servir à l'administration. Par exemple, pour lister toutes les bases de données du serveur avec leurs tailles, on passe par *master*[1] :

```
EXECUTE master.sys.sp_MSforeachdb 'USE [?]; EXEC sp_spaceused'
```

ou avec leurs tailles et dates de backup :

```
select *
FROM master.sys.databases db
LEFT OUTER JOIN msdb.dbo.backupset b ON db.name = b.database_name
```

On peut aussi voir les paramètres de configuration de SQL Server :

```
SELECT * from master.sys.configurations order by NAME
```

Création

Pour créer une base de données il faut définir le fichier dans lequel elle sera stockée :

```
CREATE DATABASE [MaBase] ON  PRIMARY
( NAME = N'MaBase', FILENAME = N'D:\DATASQL\MSSQL10.MSSQLSERVER\MSSQL\DATA\MaBase.mdf' ,
 LOG ON
( NAME = N'MaBase_log', FILENAME = N'D:\DATASQL\MSSQL10.MSSQLSERVER\MSSQL\DATA\MaBase_log
GO
```

Lecture

Ensuite pour la sélectionner, il faut soit le faire en début de script :

```
USE MaBase;
SELECT * FROM MaTable;
```

Soit appeler tous les objets par leur chemin absolu, ex :

```
SELECT * FROM [MaBase].[dbo].[MaTable];
```

Références

1. http://weblogs.sqlteam.com/joew/archive/2008/08/27/60700.aspx

Gestion des utilisateurs

Introduction

On distingue les comptes de connexions au serveur, de ceux des utilisateurs stockés dans chaque base et qui en contiennent les permissions.

On peut assigner aux premiers des "Rôles du serveur" et aux seconds des "Rôles" propres à leur base de données.

Plusieurs comptes utilisateurs peuvent utiliser la même connexion au serveur.

SSMS

Connexions

Si certaines connexions pourront lancer *Microsoft SQL Server* depuis le serveur avec des droits limités sur certaines bases, il est plutôt prévu de les faire lire la base via des requêtes HTTP par l'intermédiaire de langages tiers comme PHP ou Java.

Si toutefois ils ont besoin de lire certaines tables ou d'exécuter des procédures stockées en ODBC il peut être nécessaire de leur donner les droits :

- Menu "Sécurité" du serveur (en bas à gauche pas défaut, ne pas prendre le menu "Sécurité" d'une base).
- Connexions.
- Clic droit, "Nouvelle connexion...".
- Dans "Général", indiquer le nom du compte et la base par défaut.
- Dans "Rôles su serveur", cocher les droits demandés.

Par ailleurs, cette interface a été critiquée car contrairement à phpMyAdmin par exemple, elle ne permet ni de copier des bases ou des tables, ni d'insérer des lignes dans des tables de plus de 200 lignes, ni de rechercher des champs ou des valeurs (le code pour rechercher est publié dans les chapitres suivants).

Utilisateurs

Dans SSMS, les utilisateurs sont visibles dans le menu "Sécurité de chaque base de données.

Connections en cours

Dans SSMS, un clic droit sur une base permet d'afficher l'option *Moniteur d'activité*. Ce menu affiche en temps réel les connexions et performances de la base. Il est donc possible d'y forcer l'arrêt des connexions, par exemple avant de supprimer une base (car son utilisation empêche toute suppression).

SQL

Les connexions et utilisateurs peuvent aussi être gérés en SQL :

```
CREATE LOGIN Connexion1 WITH PASSWORD = 'test'
USE MaBase;
CREATE USER LectureSeule FOR LOGIN Connexion1;
exec sp_addrolemember 'public', 'LectureSeule'
```

Pour le réactiver :

```
ALTER LOGIN Connexion1 ENABLE
```

Pour le débloquer :

```
ALTER LOGIN Connexion1 WITH PASSWORD = 'test' UNLOCK
```

Inventaire des connexions :

```
SELECT loginname FROM master.dbo.syslogins
```

La liste des connexions en cours est listable par la commande `sp_who`[1].

Inventaire des utilisateurs de la base courante :

```
SELECT name FROM sys.database_principals where (type='S' or type = 'U')
```

Références

1. https://msdn.microsoft.com/fr-fr/library/ms174313%28v=SQL.120%29.aspx

Variables

Déclaration et affectation

Tout nom de variable commence par un arobase.

- Opération sur des entiers (***int****eger*) :

```
declare @i int
set @i = 5

declare @j int
set @j = 6

print @i+@j -- affiche 11
```

- Opération sur des caractères (***char****acter*) :

```
declare @k char
set @k = '5'

declare @l char
set @l = '6'

print @k+@l -- affiche 56
```

Types

Les types de variables possibles sont les mêmes que ceux des champs des tables[1] :

Caractères

Ceux qui commencent par "n" sont au format Unicode.

char, nchar, nvarchar, ntext, text, varchar.

Pour gagner un peu de place en mémoire, il est possible de limiter leurs tailles à un certain nombre de caractères. Ex :

```
varchar(255)
```

La taille maximale pour un *varchar* (*variable of characters*) est de 2 Go[2] :

```
varchar(MAX)
```

Nombres

decimal, int (tinyint, smallint, bigint), float, money, numeric, real, smallmoney.

Dates

date, *datetime*, *datetime2*, *datetimeoffset*, *smalldatetime*, *time*.

Types personnalisés

En plus des types natifs, il est possible de créer ses propres types de données avec `CREATE TYPE`.

Détermination du type

La fonction `SQL_VARIANT_PROPERTY` renvoie le type d'un champ donné[3]. Exemple :

```
SELECT SQL_VARIANT_PROPERTY(Champ1, 'BaseType')
FROM table1
```

Références

1. https://msdn.microsoft.com/fr-fr/library/ms187752.aspx
2. https://msdn.microsoft.com/fr-fr/library/ms176089.aspx
3. https://msdn.microsoft.com/en-us/library/ms178550.aspx

Tables

Introduction

Les langage de définition et de manipulation de données (LDD et LMD) respectent la norme SQL-86. Toutefois, en plus des requêtes `SELECT, UPDATE, INSERT` on trouve `MERGE` depuis la version 2008[1].

Créer une table

Dans SSMS, un clic droit sur le dossier "Tables" d'une base permet d'en ajouter.

Sinon en SQL il faut taper[2] :

```
CREATE TABLE [dbo].[table1] (
    [Nom] [varchar](250) NULL,
    [Prénom] [varchar](250) NULL,
    [identifiant] [int] IDENTITY(1,1) NOT NULL)
```

Un clic droit sur une table existante permet au choix de :

1. Modifier sa structure (ajouter une colonne, modifier un type).
2. Sélectionner ses 1 000 premiers enregistrements.
3. Éditer ses 200 premiers.

Pour sélectionner d'autre fractions de la table, utiliser `TOP` :

```
SELECT TOP 100 * FROM table1                   -- Les 100 premiers
SELECT TOP 100 * FROM table1 ORDER BY id DESC -- Les 100 derniers
SELECT TOP (10) PERCENT * FROM table1          -- Les 10 premiers pourcents
```

Remplir une table

Remplissage des premières colonnes[3] :

```
INSERT INTO table1 VALUES ('Doe', 'Jane', 1), ('Doe', 'John', 2)
```

Pour certaines colonnes ciblées il faut préciser les champs. Par exemple en ne remplissant que le prénom, le nom de famille sera nul :

```
INSERT INTO table1 (Prénom, identifiant) VALUES ('Jane', 3)
```

Depuis une autre table :

```
INSERT INTO table1 (Prénom, identifiant)
SELECT Prénom, ID FROM table2
```

Mise à jour :

```
UPDATE table1
SET Prénom = 'Janet'
WHERE ID = 3
```

```
UPDATE table1
SET Prénom = t2.Prénom, Nom = t2.Nom
FROM table1 t1
INNER JOIN table2 t2 on t1.ID = t2.ID_t1
```

Créer un index

L'abréviation PK du logiciel signifie "primary key" (clé primaire).

Pour créer une clé étrangère, dérouler la table, dans le menu *Clés*, clic droit, *nouvelle clé étrangère*..., la liste de toutes les clés étrangères de la table apparait dans une petite fenêtre (baptisées par défaut "FK_..." pour "foreign key" : clé étrangère).

Dans *Général*, *Spécification de tables et colonnes*, cliquer sur "..." pour sélectionner la table et son champ à lier.

Si ensuite l'erreur suivante survient :

> Les colonnes de la table ne correspondent pas à une clé primaire existante ou à une contrainte UNIQUE

Il faut définir une contrainte d'unicité[4].

Ajouter un identifiant unique

Normalement chaque table doit posséder au moins un identifiant unique (clé primaire). Or, il est impossible de modifier une colonne existante pour lui attribuer la propriété `AUTOINCREMENT` nécessaire à une telle clé.

Il faut donc en ajouter une :

```
ALTER TABLE table1 ADD id int NOT NULL IDENTITY (1,1) PRIMARY KEY
```

Copier une table

La sélection ci-dessous clone une table avec les mêmes tailles de champs :

```
SELECT * INTO table2 FROM table1
```

Sachant que la table `spt_values` de la base système `master` fournit déjà des numéros séquentiels via son champ `number`, il devient possible de générer des tables préremplies avec le compteur :

```
SELECT DISTINCT number
FROM master.dbo.spt_values
WHERE number BETWEEN 2 AND 10
```

D'où :

```
SELECT DISTINCT 'Ligne ' + convert(varchar, number, 112) as No into #TableVierge
FROM master.dbo.spt_values
WHERE number BETWEEN 2 AND 10

SELECT * from #TableVierge
```

```
No
Ligne 10
Ligne 2
Ligne 3
Ligne 4
Ligne 5
Ligne 6
Ligne 7
Ligne 8
Ligne 9
```

Importer une table

Soit un tableau (Excel ou Calc) converti par exemple en CSV encodé en PC DOS, pour l'importer en tant que nouvelle table[5] :

```
CREATE TABLE Tableau_to_Table (
  [Champ1] [varchar](500) NULL,
  [Champ2] [varchar](500) NULL,
  [Champ3] [varchar](500) NULL
)
GO
BULK INSERT Tableau_to_Table
FROM 'C:\Users\superadmin\Desktop\Tableau1.csv'
WITH (
  FIELDTERMINATOR = ';',
  ROWTERMINATOR = '\n'
)
GO
-- Affiche le résultat
SELECT * from Tableau_to_Table
GO
```

Attention !

Si la table possède un ID en autoincrémentation, le contenu du CSV s'ajoutera à la suite. Pour éviter cela, utiliser `DBCC CHECKIDENT`[6].

Supprimer une table

Pour supprimer toute la table (données et structure) :

```
DROP TABLE table1
```

Pour tronquer une table, c'est-à-dire ne conserver que les en-têtes et types des colonnes, en retirant tous les enregistrements :

```
TRUNCATE TABLE table1
--ou
DELETE table1
```

Pour supprimer certaines lignes d'une table :

```
DELETE table1 WHERE Condition
```

> **Remarque :** en ajoutant `OUTPUT deleted.*` avant le `WHERE`, on obtient le contenu supprimé au lieu du nombre de lignes supprimées.

Rechercher une table

Pour rechercher une table dont on connait le nom exact, dans toutes les bases de données du serveur :

```
sp_MSforeachdb 'USE ?
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N''[MaTableConnue]'') AND OB
BEGIN
  PRINT ''Table trouvée dans la base : ?''
END'
```

Rechercher dans toutes les tables

SSMS 10 ne propose pas de fonction de recherche de table, champ ou valeur, comme on en trouve dans phpMyAdmin pour MySQL par exemple.

Recherche d'une table

Ce script parcourt chaque base de données pour y récupérer les tables dont les noms contiennent une chaine de caractères spécifiée (à la fin).

```
ALTER Proc FindTable
@TableName nVarchar(50)
As
/*
Purpose : Search for a Table in all databases
Author : Sandesh Segu
Date : 17th July 2009
Version : 1.0
More Scripts  : http://sanssql.blogspot.com
*/
ALTER Table #temp (DatabaseName varchar(50),SchemaName varchar(50),TableName varchar(50))

Declare @SQL Varchar(500)
Set @SQL='Use [?] ;
if exists(Select name from sys.tables where name like '''+@TableName+''')
insert into #temp
Select ''?'' AS DatabaseName ,SS.Name AS SchemaName ,ST.Name AS TableName from sys.tables
where ST.Schema_ID=SS.Schema_ID and ST.name like '''+@TableName+''''

EXEC sp_msforeachdb @SQL
```

```
Select * from #temp

Drop table #temp
GO

EXEC FindTable '%Chaine à rechercher%'
```

Recherche d'une valeur

La recherche d'une valeur de champ dans toutes les tables prend un certain temps[7] :

```
CREATE TABLE #result(
  id      INT IDENTITY,
  tblName VARCHAR(255),
  colName VARCHAR(255),
  qtRows  INT
)
go

DECLARE @toLookFor VARCHAR(255)
SET @toLookFor = '%Chaine à rechercher%'

DECLARE cCursor CURSOR LOCAL FAST_FORWARD FOR
SELECT
  '[' + usr.name + '].[' + tbl.name + ']' AS tblName,
  '[' + col.name + ']' AS colName,
  LOWER(typ.name) AS typName
FROM
  sysobjects tbl
    INNER JOIN(
      syscolumns col
        INNER JOIN systypes typ
        ON typ.xtype = col.xtype
    )
    ON col.id = tbl.id
    --
    LEFT OUTER JOIN sysusers usr
    ON usr.uid = tbl.uid

WHERE tbl.xtype = 'U'
  AND LOWER(typ.name) IN(
        'char', 'nchar',
        'varchar', 'nvarchar',
        'text', 'ntext'
      )
ORDER BY tbl.name, col.colorder
--
DECLARE @tblName VARCHAR(255)
DECLARE @colName VARCHAR(255)
DECLARE @typName VARCHAR(255)

DECLARE @sql  NVARCHAR(4000)
DECLARE @crlf CHAR(2)

SET @crlf = CHAR(13) + CHAR(10)

OPEN cCursor
FETCH cCursor
INTO @tblName, @colName, @typName

WHILE @@fetch_status = 0
BEGIN
  IF @typName IN('text', 'ntext')
  BEGIN
```

```
    SET @sql = ''
    SET @sql = @sql + 'INSERT INTO #result(tblName, colName, qtRows)' + @crlf
    SET @sql = @sql + 'SELECT @tblName, @colName, COUNT(*)' + @crlf
    SET @sql = @sql + 'FROM ' + @tblName + @crlf
    SET @sql = @sql + 'WHERE PATINDEX(''%'' + @toLookFor + ''%'', ' + @colName + ') > 0'
  END
  ELSE
  BEGIN
    SET @sql = ''
    SET @sql = @sql + 'INSERT INTO #result(tblName, colName, qtRows)' + @crlf
    SET @sql = @sql + 'SELECT @tblName, @colName, COUNT(*)' + @crlf
    SET @sql = @sql + 'FROM ' + @tblName + @crlf
    SET @sql = @sql + 'WHERE ' + @colName + ' LIKE ''%'' + @toLookFor + ''%''' + @crlf
  END

  EXECUTE sp_executesql
            @sql,
            N'@tblName varchar(255), @colName varchar(255), @toLookFor varchar(255)',
            @tblName, @colName, @toLookFor

  FETCH cCursor
  INTO @tblName, @colName, @typName
END

SELECT *
FROM #result
WHERE qtRows > 0
ORDER BY id
GO

DROP TABLE #result
go
```

Créer une vue

Pour créer une vue :

```
CREATE VIEW vue1
AS
SELECT DISTINCT champ1
FROM table1;
```

> **Remarque :** dans SSMS, l'affichage d'une telle vue n'est pas plus rapide que la requête l'ayant générée, sauf si elle a un index cluster[8].

Dans ce cas, la différence de temps d'affichage est très significative :

```
CREATE VIEW vue1 WITH SCHEMABINDING
AS
SELECT COUNT_BIG(*) as Compte, champ1
FROM dbo.table1
group by champ1;

SELECT champ1 FROM vue1 WITH (NOEXPAND);
```

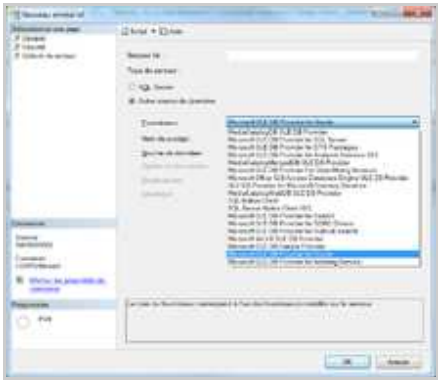
Références

1. https://msdn.microsoft.com/fr-fr/library/bb510625%28v=sql.120%29.aspx

2. https://msdn.microsoft.com/en-us/library/ms174979.aspx
3. https://msdn.microsoft.com/en-us/library/ms174335.aspx
4. http://office.microsoft.com/fr-fr/help/les-colonnes-de-la-table-0s-ne-correspondent-pas-a-une-cle-primaire-existante-ou-a-une-contrainte-unique-HP003083867.aspx
5. http://msdn.microsoft.com/fr-fr/library/ms188365.aspx
6. https://msdn.microsoft.com/fr-fr/library/ms176057%28v=sql.120%29.aspx?f=255&MSPPError=-2147217396
7. http://stackoverflow.com/questions/591853/search-for-a-string-in-an-all-the-tables-rows-and-columns-of-a-db
8. https://technet.microsoft.com/en-us/library/dd171921%28v=sql.100%29.aspx

Procédures stockées

Introduction



Ajout d'un serveur lié, il peut être de plusieurs types dont Oracle Database.

Les procédures stockées sont des ensembles de requêtes SQL enregistrés dans les bases de données. Dans SSMS, on les trouve dans le menu du même nom à côté de celui des tables.

En effet, d'un point de vue de l'architecture logicielle d'une application, comme les longues suites de requêtes avec des structures de contrôles sont propres à leur SGBD, il est préférable de les grouper avec les données, pour permettre de passer d'un SGBD à l'autre sans redévelopper le module de formulaire d'interaction avec l'utilisateur (ex : un site Web peut ainsi passer de MySQL à MSSQL sans être repris intégralement, car il invoque une procédure stockée de même nom, avec les mêmes entrées et sorties, dans les deux SGBD).

Les procédures stockées servent généralement à manipuler les tables de la base où elles se trouvent, mais peuvent également interagir avec celles d'autres bases (dont les noms sont placés en préfixe) du même serveur, ou de serveurs liés. Pour créer un serveur lié dans SSMS, se rendre dans le menu "Objets serveur", puis "Serveurs liés", et remplir le compte à utiliser pour s'y connecter (ou utiliser `sp_addlinkedserver`[1] où "sp" signifie "stored procedure").

Exemple de jointure entre deux serveurs :

```
select *
from table1 t1
inner join [serveur2].[base2].[dbo].[table2] t2 on t2.id = t1.t2_id
```

Syntaxe

Le langage T-SQL de Microsoft contient quelques améliorations par rapport à la norme SQL :

- Les guillemets ont un rôle différent des apostrophes qui servent à créer des chaines de caractères. Pour les utiliser comme tels il faut donc lancer `SET QUOTED_IDENTIFIER ON`.
- Dans SSMS, une requête SQL peut être exécutée de trois façons :

1. Soit directement dans une fenêtre blanche apparaissant quand on clique sur "Nouvelle requête". On peut en sauvegarder le contenu en .sql, pour pouvoir la rouvrir plus tard.
2. Soit en stockant la requête dans une variable, avant d'exécuter cette dernière avec `sp_executesql`[2]. Ce qui a l'avantage pouvoir y incorporer des variables (ex : nom d'une base de données), mais l'inconvénient de supprimer la coloration syntaxique, l'autocomplétion (IntelliSense[3]) et le débogage SSMS. Ex :

```
DECLARE @Requete1 NVARCHAR(MAX)
DECLARE @MaTable1 NVARCHAR(MAX)
SET @MaTable1 =
SET @Requete1 =  'SELECT * FROM ' + @MaTable1
EXECUTE sp_executesql @Requete1
```

3. Soit en exécutant une procédure stockée dans une base de données (à côté des tables), dans laquelle on a

enregistré une requête. Ex :

```
EXEC [MaBase1].[dbo].[MaProcédure1]
```

Cet appel peut être suivi d'arguments, comme une procédure ou fonction en programmation impérative.

En effet, on en distingue deux sortes de variables dans les procédures stockées :

1. Si elles le sont avec le mot *Declare*, elles sont privées.
2. Sans ce mot, elles représentent les variables externes de la procédures, à préciser lors de son exécution :

```
@DateDebut varchar(8)       --Variable publique obligatoire comme argument
@DateFin varchar(8) = null  --Variable publique facultative
if @DateFin is null set @DateFin=convert(varchar,@DateDebut+1,112)
Declare @Nom varchar(50) --Variable privée
```

Pour créer une nouvelle procédure stockée :

```
CREATE PROCEDURE [dbo].[MaProcédure1]
```

Pour enregistrer une procédure stockée existante, il faut exécuter :

```
ALTER PROCEDURE [dbo].[MaProcédure1]
```

Idéalement cette instruction sera présente au début de la procédure stockée suivie de `AS`, dont exécution aura donc pour effet de l'enregistrer (et pas d'en exécuter le contenu). Pour obtenir son résultat, il faut effectuer un clic droit dessus, puis choisir "Exécuter la procédure stockée..." : cela génère une autre requête SQL qui s'ouvre dans un nouvel onglet au-dessus du résultat, appelant la procédure stockée avec ses paramètres.

Attention !

SSMS ne tolère pas qu'on sauvegarde une procédure stockée avec des erreurs de compilation. En cas de besoin il faut donc commenter le code en erreurs, ou passer par un fichier .sql (temporaire).

Attention !

Les messages d'erreur communiquent un numéro de ligne décalé par rapport à celui numéroté par l'interface. Il faut y soustraire le nombre de lignes présentes avant le dernier `GO`.

Par la suite, ces procédures stockées peuvent ensuite être appelées dans des programmes dans des langages qui contiennent un pilote SQL Server, tels que PHP ou VB, qui en présenteront les résultats.

PRINT

Cette commande affiche des caractères (variables ou constantes) dans l'onglet *Messages*, contrairement au

SELECT qui remplit l'onglet *Résultats.*

Exemples :

```
print 'Hello World ! ' -- Affiche Hello World !

declare @n int
set @n = 5

print 'la valeur est : ' + cast(@n as varchar)
```

Conditions

IF

```
if @x=1 begin
  print 'x = 1'
end else if @x=2 begin
  print 'x = 2'
end else begin
  print 'x <> 1 et 2'
end
```

Remarque : le *begin* et le *end* peuvent être facultatifs.

CASE

```
set @Saison = case
 when @Datejour = '20110918' then 'été'
 when @Datejour = '20110922' then 'automne'
 else 'autre saison'
end
```

Pour ajouter une condition `WHERE` uniquement si une valeur est présente, il faut que dans le cas contraire, la deuxième condition soit toujours vraie (ex : Champ1 = Champ1) :

```
declare @Colonne int = null
select Champ1
from Table1
where Champ1 = case when isnull(@Colonne,'')<>'' then @Colonne else Champ1 end
```

A noter que l'exemple ci-dessus serait plus simple avec `where Champ1 = isnull(@Colonne,Champ1).`

Boucles

WHILE[4]

La boucle "while" utilise un compteur dans sa condition :

```
DECLARE @i int
WHILE @i <= 10
BEGIN
```

```
   UPDATE table1
   SET champ2 = "petit" WHERE champ1 = @i
   SET @i = @i + 1
END
```

CURSOR

Un curseur permet de traiter un jeu d'enregistrements ligne par ligne, chacun étant stocké dans les variables suivant le INTO, et réinitialisé après le NEXT[5]. Toutefois il est relativement lent est doit être remplacé par d'autres techniques quand c'est possible[6].

On peut par exemple ajouter des caractères sur certaines :

```
USE Base1
declare @Nom varchar(20)
DECLARE curseur1 CURSOR FOR SELECT Prenom FROM Table1
OPEN curseur1

/* Premier enregistrement de la sélection */
FETCH NEXT FROM curseur1 into @Nom
print 'Salut ' + @Nom

/* Traitement de tous les autres enregistrements dans une boucle */
while @@FETCH_STATUS = 0
  begin
    FETCH NEXT FROM curseur1 into @Nom
    print 'Salut ' + @Nom
  end

CLOSE curseur1;
DEALLOCATE curseur1;
```

Exécution de procédures depuis d'autres

Tout comme dans l'environnement de développement intégré Visual Basic, il existe un mode d'exécution pas à pas : en pressant F11 à chaque arrêt il est possible de suivre le lancement du programme, tout en surveillant les valeurs des variables en bas à gauche.

Les points d'arrêt sont également disponibles pour personnaliser les pas.

Toute modification de la procédure stockée pendant ce débogage s'affichera, mais ne sera pas pris en compte par le processus.

Pour exécuter une procédure stockée depuis une autre :

```
ALTER PROCEDURE [dbo].[MaProcédure1]
DECLARE @resultat int
EXEC    @resultat = [dbo].[MaProcédure2] @Parametre1;
if @resultat = 0 begin
 ...
end
```

Exceptions

Apparue avec SQL Server 2005, la gestion d'exceptions se présente ainsi :

```
-- Début de la transaction
BEGIN TRAN
  BEGIN TRY
    -- Exécution
    INSERT INTO Table1(Nom1) VALUES ('ABC')
    INSERT INTO Table1(Nom1) VALUES ('123')
    -- Soumission de la transaction
    COMMIT TRAN
  END TRY
BEGIN CATCH
 -- Annulation de la transaction si erreur
 ROLLBACK TRAN
END CATCH
```

Recherches

Pour obtenir la liste des procédures stockées contenant une chaine particulière :

```
SELECT name
FROM sysobjects syso
INNER JOIN syscomments sysc
ON syso.id = sysc.id
WHERE
(syso.xtype = 'P' or
syso.xtype = 'V')
AND
(syso.category = 0)
and text like '%Chaine à rechercher%'
group by name
```

Références

1. https://msdn.microsoft.com/fr-fr/library/ms190479.aspx
2. https://msdn.microsoft.com/en-us/library/ms188001.aspx?f=255&MSPPError=-2147217396
3. https://msdn.microsoft.com/fr-fr/library/hcw1s69b.aspx?f=255&MSPPError=-2147217396
4. http://msdn.microsoft.com/fr-fr/library/ms178642.aspx
5. http://msdn.microsoft.com/fr-fr/library/ms180169.aspx
6. http://sqlpro.developpez.com/cours/sqlserver/MSSQLServer-avoidCursor/

Fonctions

Environnement SERVERPROPERTY

La fonction suivante permet de récupérer le nom du serveur courant :

```
SELECT SERVERPROPERTY('MachineName')
```

Condition ISNULL

Cette fonction ne renvoie pas *vrai* si la variable est nulle, comme dans d'autres langages de programmation, mais plutôt son second paramètre, un substitut à NULL obligatoire. Ceci permet de lever les exceptions sur les variables nulles directement dans les requêtes, sans ajout de ligne supplémentaire.

Voici un exemple où Les NULL sont traités comme les vides :

```
select Champ1 = case when isnull(@Colonne,'')='' then '*' else @Colonne end
from Table1
```

Extrémums : MIN, MAX

Les fonctions *Min()* et *Max()* renvoient respectivement le minimum et le maximum d'une liste de champs.

```
select min(Date) from Calendrier where RDV = 'Important'
```

Conversions : CAST et CONVERT

CAST modifie le type d'une variable :

```
cast(Champ as decimal(12, 6)) -- sinon '9' > '10'
```

CONVERT modifie le type d'une variable en premier paramètre, et sa longueur en second.

```
convert(varchar, Champ1, 112)
convert(datetime, Champ2, 112)       -- sinon impossible de parcourir le calendrier (ex
```

> **Attention !**
>
> Tous les types de variable ne sont pas compatibles entre eux[1].

Exemple de problème :

```
select Date1
```

```
from Table1
where Date1 between '01/10/2013' and '31/10/2013'
```

Les dates ne sont pas forcément reconnues sans utiliser `convert`. La solution est donc de stocker les dates si possible dans le format `datetime` :

```
select Date1
from Table1
where Date1 between convert(varchar,'20131001',112) and convert(varchar,'20131031',112)
```

Si par contre la date du paragraphe ci-dessus est stockée en `varchar` avec des slashs, il devient obligatoire de la reformater pour pouvoir la comparer.

De nombreux formats de dates sont disponibles[2].

Arrondis : FLOOR, CEILING, ROUND

Pour arrondir un nombre :

- A l'inférieur : `floor(nombre)`.
- Au supérieur : `ceiling(nombre)`.
- Au nombre de chiffres précisé : `round(nombre, chiffres)`. Ex :
  `round(499, -3)` = 0.
  `round(500, -3)` = 1000.
  `round(0.45, 1)` = 0.50.
  `round(0.44, 1)` = 0.40.

> **Attention !**
>
> Pour arrondir une opération sur un entier (ex : `COUNT`), il faut le convertir sinon il s'arrondit à l'inférieur. Ex :
>
> ```
> SELECT ceiling(13 / 12) // = 0
> SELECT ceiling(convert(decimal(12, 6), 13) / 12) // = 1
> ```

Troncatures : LEFT, RIGHT, et SUBSTRING

Permettent de découper des chaines de caractères selon les positions de leurs caractères[3].

```
select substring('13/10/2013 00:09:19', 7, 4) -- renvoie les quatre caractères à partir
```

Par exemple dans le cas de la date avec slashs vue dans le paragraphe précédent :

```
select Date1
from Table1
where right(Date1, 4) + substring(Date1, 4, 2) + left(Date1, 2) between convert(varchar,
```

Recherche CHARINDEX

Pour rechercher la position d'un mot dans une chaine de caractères[4] :

```
select CHARINDEX ('table2', 'table1 table2 table3');
-- égal 8
```

Manipulations : REPLACE et STUFF

Permettent de remplacer des caractères dans une chaine selon leur valeur[5] : rechercher et remplacer.

Par exemple pour mettre à jour le chemin d'un répertoire renommé[6] :

```
update Table1
SET Champ1 = replace(Champ1,'\Ancien chemin\','\Nouveau chemin\')
where Champ1 like '%\Ancien chemin\%'
```

Dates

Format date

La fonction `GETDATE` est utilisée pour la date courante. Pour obtenir plutôt la date d'un jour donné au format date, il faut utiliser `CONVERT` :

```
select convert(smalldatetime, '2016-01-02', 121)
```

Découpage

La fonction `DATEPART` extrait une partie de date sans avoir besoin de son emplacement[7].

Toutefois, trois fonctions permettent d'accélérer l'écriture de ce type d'extractions :

```
-- Jour
select day(getdate())
-- Mois
select month(getdate())
-- Année
select year(getdate())
-- Année précédente
select getdate(), 'Année précédente : ' + str(year(getdate()) - 1)
```

Noms

La fonction `DATENAME()`[8] permet d'obtenir les noms des éléments d'une date. Ex :

```
select datename(month, '20160712')   -- affiche "juillet"
select datename(weekday, '20160712') -- affiche "mardi"
```

Addition et soustraction de jours

Voici deux fonctions de manipulation de dates[9] :

- `DATEDIFF` calcule l'intervalle entre deux dates[10].
- `DATEADD` retourne la date issue d'une autre plus un intervalle[11].

```
-- Dernier jour du mois précédent
SELECT DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,GETDATE()),0))
-- Dernier jour du mois courant
SELECT DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,GETDATE())+1,0))
-- Dernier jour du mois prochain
SELECT DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,GETDATE())+2,0))
```

Exemple :

```
SELECT DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,'20150101'),0)) as date
```

donne :

```
date
2014-12-31 23:59:59.000
```

Personnalisée

On peut aussi créer ses propres fonctions avec `CREATE FUNCTION`[12], qui sont stockées dans un menu séparé à côté des tables et des procédures stockées.

```
CREATE FUNCTION MaFonction1(@Paramètre1 varchar) returns varchar AS
BEGIN
  return 'Hello ' + @Paramètre1
END
```

Triggers

Les triggers sont des scripts qui se déclenchent selon certains évènements, et sont créés avec `CREATE TRIGGER`[13]. Dans SSMS, on les trouvent dans le menu *Déclencheurs de base de données*.

```
CREATE TRIGGER MonTrigger1
ON ALL SERVER
FOR CREATE_DATABASE
AS
BEGIN
  PRINT 'Une base a été créée.'
END
```

Références

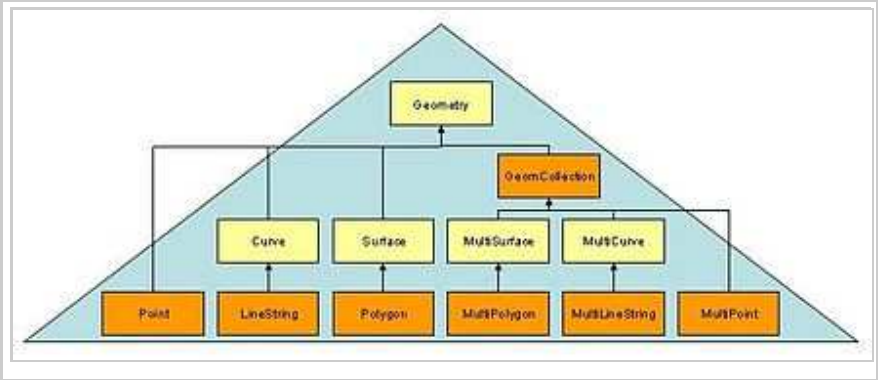
1. man CONVERT (http://msdn.microsoft.com/fr-fr/library/ms187928.aspx) [archive]

2. http://stackoverflow.com/questions/74385/how-to-convert-datetime-to-varchar
3. man SUBSTRING (http://msdn.microsoft.com/fr-fr/library/ms187748.aspx) [archive]
4. https://msdn.microsoft.com/fr-fr/library/ms186323.aspx
5. man STUFF (http://msdn.microsoft.com/fr-fr/library/ms188043.aspx) [archive]
6. man REPLACE (http://msdn.microsoft.com/fr-fr/library/ms186862.aspx) [archive]
7. man DATEPART (http://msdn.microsoft.com/fr-fr/library/ms174420.aspx) [archive]
8. https://msdn.microsoft.com/fr-fr/library/ms174395.aspx
9. http://blog.sqlauthority.com/2007/08/18/sql-server-find-last-day-of-any-month-current-previous-next/
10. man DATEDIFF (http://msdn.microsoft.com/fr-fr/library/ms189794.aspx) [archive]
11. man DATEADD (http://msdn.microsoft.com/fr-fr/library/ms186819.aspx) [archive]
12. https://msdn.microsoft.com/fr-fr/library/ms186755(v=sql.120).aspx
13. https://msdn.microsoft.com/fr-fr/library/ms189799(v=sql.100).aspx

Bases de données spatiales

Principe

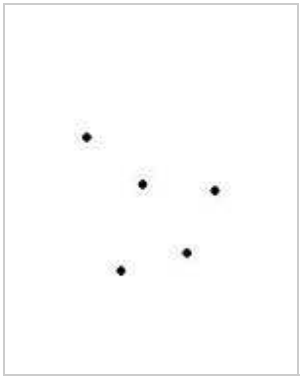
Lors du typage des champs, certains représentent des objets graphiques, et sont donc considérés comme étant de catégorie "Spatial" (base de données spatiales). Par conséquent, ils se manipulent par des requêtes différentes que pour le texte.



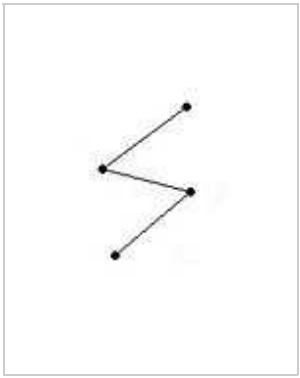
On distingue cinq types de champs[1] :

1. CircularString
2. CompoundCurve
3. LineString
4. Point
5. Polygon

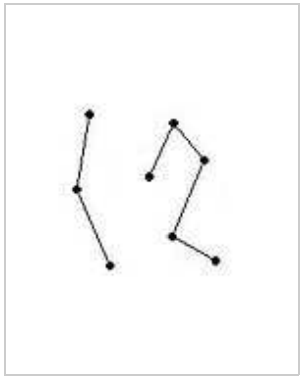
Point



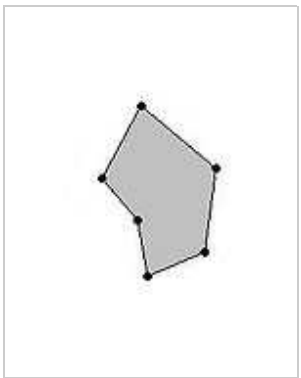
MultiPoint



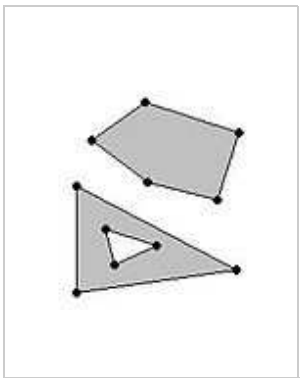
LineString



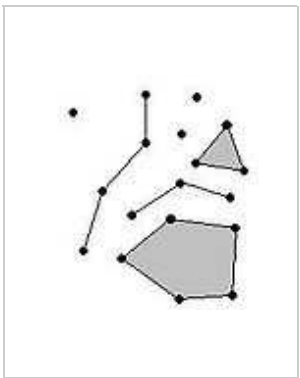
MultiLineString



Polygon



MultiPolygon



GeometryCollection

Et 11 types de relations entre eux[2] :

1. STEquals
2. STDisjoint
3. STIntersects

4. STTouches
5. STOverlaps
6. STCrosses
7. STWithin
8. STContains
9. STOverlaps
10. STRelate
11. STDistance

Requêtes

```
CREATE TABLE Districts
( DistrictId int IDENTITY (1,1),
DistrictName nvarchar(20),
DistrictGeo geometry);
GO

CREATE TABLE Streets
( StreetId int IDENTITY (1,1),
StreetName nvarchar(20),
StreetGeo geometry);
GO

INSERT INTO Districts (DistrictName, DistrictGeo)
VALUES ('Downtown',
geometry::STGeomFromText
('POLYGON ((0 0, 150 0, 150 150, 0 150, 0 0))', 0));

INSERT INTO Districts (DistrictName, DistrictGeo)
VALUES ('Green Park',
geometry::STGeomFromText
('POLYGON ((300 0, 150 0, 150 150, 300 150, 300 0))', 0));

INSERT INTO Districts (DistrictName, DistrictGeo)
VALUES ('Harborside',
geometry::STGeomFromText
('POLYGON ((150 0, 300 0, 300 300, 150 300, 150 0))', 0));

INSERT INTO Streets (StreetName, StreetGeo)
VALUES ('First Avenue',
geometry::STGeomFromText
('LINESTRING (100 100, 20 180, 180 180)', 0))
GO

INSERT INTO Streets (StreetName, StreetGeo)
VALUES ('Mercator Street',
geometry::STGeomFromText
('LINESTRING (300 300, 300 150, 50 51)', 0))
GO
```

```
SELECT ...
```

Cette section est vide, pas assez détaillée ou incomplète.

Références

1. https://msdn.microsoft.com/fr-fr/library/bb964711%28v=sql.120%29.aspx?f=255&MSPPError=-2147217396
2. https://msdn.microsoft.com/fr-FR/library/bb895270%28v=sql.120%29.aspx

Hints

Optimisation de requêtes

Les hints ou *indicateurs* permettent d'optimiser les transactions pour arriver au même résultat plus rapidement avec moins de ressources.

Pour comparer ceux-ci, il suffit de lancer dans le menu *Requête, Afficher le plan d'exécution estimé*.

Lors d'une exécution via SQL Studio, le bouton "include le plan d'exécution réel" permet de voir si l'ordre des opérations est le plus judicieux (généralement on procède par projection, puis sélection et jointure). Ce plan peut être enregistré en XML, au format .sqlplan.

Les hints sont stipulés à la fin de la requête, avec les clauses `WITH` ou `OPTION` entre parenthèses[1].

Index

Il existe plusieurs types d'indexation des tables sur MSSQL[2] :

- index unique : index appliqué sur une clé candidate. On le crée avec la clause `CREATE UNIQUE INDEX` ;
- index non-cluster : index par défaut si aucun n'est précisé (`CREATE INDEX` = `CREATE NONCLUSTERED INDEX`) ;
- index cluster : particularité de Microsoft SQL Server[3] qui stocke les données dans les feuilles de l'arbre. L'ordre physique correspond donc à l'ordre logique des enregistrements ;
- index XML primaire : pour les balises XML ;
- index spatial : pour les bases de données spatiales ;
- index columnstore.

Pour savoir quel sont les index d'une table :

```
sp_helpindex  table1
```

Création

Par convention on nommera les index avec un préfixe "IX_" :

```
CREATE INDEX IX_Date
ON MaBase.table1(Champ_Date);
```

Utilisation

Pour permettre à l'optimiseur de requêtes un élagage rapide des enregistrements à ne pas parcourir lors d'une sélection, il faut utiliser le ou les index les plus appropriés par rapport aux conditions (`WHERE`) :

```
SELECT *
FROM table1 WITH (INDEX(IX_Date))
WHERE Champ_Date between '20150101' and '20150131'
```

Suppression

```
DROP INDEX MaBase.table1.IX_Date
```

Group by

GROUP BY avec ROLLUP, CUBE et GROUPING SETS[4].


Cette section est vide, pas assez détaillée ou incomplète.

Audits

Prévoir un audit trimestriel pour déterminer les requêtes les plus consommatrices[5].

Références

1. http://msdn.microsoft.com/fr-fr/library/ms187713%28v=sql.100%29.aspx
2. https://msdn.microsoft.com/fr-fr/library/ms188783.aspx
3. https://msdn.microsoft.com/fr-fr/library/ms190457.aspx
4. https://technet.microsoft.com/fr-fr/library/bb522495%28v=sql.105%29.aspx
5. http://www.dbta.com/Editorial/Trends-and-Applications/Essential-Tips-on-SQL-Server-Database-Performance-108768.aspx

Importer et exporter

Interfaces graphiques

Par défaut dans SSMS, le bouton *Résultats dans des grilles* est enfoncé. En cliquant sur *Résultats dans un fichier*, il devient possible d'exporter le contenu de tables dans un fichier .rpt avec une requête SQL.

Pour transformer une base MS-SQL en MySQL il existe MySQL Workbench[1]. Attention car la syntaxe des procédures stockées est différente[2].

xp_cmdShell

`xp_cmdShell` permet de d'interagir avec le système de fichier, en manipulant le shell du système d'exploitation.

Créer un fichier texte

```
EXECUTE master.dbo.xp_cmdShell 'echo Hello World! > C:\Test.txt'
-- ou en abrégeant :
xp_cmdShell 'echo Hello World! > C:\Test.txt'
```

Pour le lire, il suffit de le charger dans une table avec `BULK INSERT`.

Copier un fichier

```
xp_cmdshell 'copy C:\Test.txt C:\Test2.txt'
```

Supprimer un fichier

```
xp_cmdshell 'del C:\Test2.txt'
```

OPENROWSET

Pour archiver vers d'autres formats que ceux de la base de données, il existe la fonction *OPENROWSET*[3]. Elle permet d'importer ou d'exporter des données aux formats MS-Access ou MS-Excel[4].

Soit le fichier C:\Test_OPENROWSET.xls existant, avec une feuille nommée "Feuil1", dont les en-têtes de colonnes sont "Nom" et "Prénom".

Insertions dans un tableur

```
insert into OPENROWSET('Microsoft.Jet.OLEDB.4.0', 'Excel 8.0;Database=C:\Test_OPENROWSET.
SELECT Nom, Prénom
FROM table1
```

> **Remarque :** si le fichier XLS(X) existe, il sera rempli à la suite avec le format de sa dernière ligne. Pour forcer les cellules en format texte (pour éviter les arrondis), il faut donc faire précéder leurs valeurs d'un

apostrophe.

Sélections dans un tableur

CSV

On définit le dossier puis le fichier :

```
SELECT * FROM OPENROWSET('MICROSOFT.JET.OLEDB.4.0','Text;Database=C:\;','SELECT * FROM [T
```

XLS

Pour lire un XLS ou ou l'importer dans une table, on définit le fichier puis la feuille :

```
select * from OPENROWSET('Microsoft.Jet.OLEDB.4.0', 'Excel 8.0;Database=C:\Test_OPENROWSE
```

Plusieurs options peuvent aussi être placées après le nom du fichier. Elles n'impactent pas le mode écriture, seulement le mode lecture.

- "HDR" (comme *header*) : les en-têtes de colonnes sont sur la première ligne (comportement par défaut) "HDR=YES". Si "HDR=NO", les colonnes sélectionnées s'appellent alors "F1", "F2", "F3"...
- "IMEX" (pour *ImportMixedTypes*)[5] :
  - "=0" : Export mode, Excel devine les types des champs.
  - "=1" : Import mode, les champs sont tous convertis en texte.
  - "=2" : Linked mode.
- Types des données[6] :
  1. "DT_BOOL" : booléen.
  2. "DT_CY" : devise.
  3. "DT_DATE" : date et heure.
  4. "DT_NTEXT" : texte.
  5. "DT_R8" : numérique.
  6. "DT_WSTR" : chaîne de caractères.

On peut aussi créer un serveur lié pour l'occasion[7] :

```
EXEC sp_addlinkedserver
    @server = 'ExcelServer1',
    @srvproduct = 'Excel',
    @provider = 'Microsoft.Jet.OLEDB.4.0',
    @datasrc = 'C:\Test_OPENROWSET.xls',
    @provstr = 'Excel 8.0;IMEX=1;HDR=YES;'
GO
SELECT * FROM ExcelServer1...[Sheet1$]
GO
SELECT * FROM OPENROWSET(ExcelServer1, 'SELECT * FROM [Sheet1$]')
```

XLSX

Pour un XLSX c'est un autre pilote :

```
select * from OPENROWSET('Microsoft.ACE.OLEDB.12.0', 'Excel 12.0;Database=C:\Test_OPENROW
```

S'il faut enregistrer le pilote, l'installer depuis : https://www.microsoft.com/fr-FR/download/details.aspx?id=23734.

Si ce pilote XLSX demande une activation[8], il faut configurer :

```
sp_configure 'Show Advanced Options', 1;
RECONFIGURE;
GO
sp_configure 'Ad Hoc Distributed Queries'; -- affiche l'état avant configuration
GO
sp_configure 'Ad Hoc Distributed Queries', 1;
RECONFIGURE;
GO
sp_configure 'Ad Hoc Distributed Queries'; -- affiche l'état après configuration
GO
```

Modification d'un tableur

La fonction `OPENROWSET` de SQL Server 2008 ne permet pas de modifier les propriétés des cellules d'un tableur, pour se faire se reporter au paragraphe sur `sp_OACreate`. Dans tous les cas, il peut tout à fait mettre à jour ses valeurs comme si elles étaient dans des tables :

```
UPDATE OPENROWSET('Microsoft.Jet.OLEDB.4.0','Excel 8.0;Database=C:\Test_OPENROWSET.xls;HD
SET F1 = '2'
WHERE F1 = '1'
```

> **Remarque :** la commande `DELETE` ne fonctionne pas sur des lignes Excel.

BCP

bcp est un utilitaire d'importation et exportation de données avec des fichiers XML[9].

sp_OACreate

`master.dbo.sp_OAMethod` est une procédure stockée étendue permettant de manipuler des fichiers et des dossiers[10]. Elle n'est pas activée par défaut sous SQL Server 2008, il faut donc le faire ainsi :

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures'; -- affiche l'état avant configuration
GO
sp_configure 'Ole Automation Procedures', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures'; -- affiche l'état après configuration
```

Voici un exemple de création puis de modification de fichier XLS (les numéros des couleurs sont les mêmes

qu'en VBA[11]) :

```
DECLARE @r int, -- résultats des commandes
    @FileName varchar(512),
    @Excel int,
    @WorkBooks int,
    @WorkBook int,
    @WorkSheet int,
    @Cells int

-- 1) Création
SET @FileName = 'C:\Test_sp_OACreate.xls'
EXEC @r = sp_OACreate 'Excel.Application', @Excel output
IF @r=0 EXEC @r = sp_OAMethod @Excel, 'Workbooks', @WorkBooks OUTPUT
IF @r=0 EXEC @r = sp_OAMethod @WorkBooks, 'Add', @WorkBook OUTPUT, -4167
IF @r=0 EXEC @r = sp_OAMethod @WorkBook, 'Worksheets(1)', @WorkSheet output, 2
IF @r=0 EXEC @r = sp_OAMethod @WorkSheet, 'Activate'
IF @r=0 EXEC @r = sp_OASetProperty @WorkSheet, 'Name', 'Reporting'
IF @r=0 EXEC @r = sp_OASetProperty @WorkSheet, 'Cells(1,3).Value', 'Hello World!'
IF @r=0 EXEC @r = sp_OAMethod @WorkBook, 'SaveAs', NULL, @FileName
IF @r=0 EXEC @r = sp_OAMethod @WorkBook, 'Close'

-- 2) Une fois le fichier fermé on peut le remplir ici avec OPENROWSET

-- 3) Retouches
IF @r=0 EXEC @r = sp_OAMethod @Excel, 'WorkBooks.Open', @WorkBook output, @FileName
IF @r=0 EXEC @r = sp_OAMethod @WorkBook, 'Worksheets(1)', @WorkSheet output, 2
IF @r=0 EXEC @r = sp_OAMethod @WorkSheet, 'Activate'

EXEC @r = sp_OASetProperty @WorkSheet, 'Range("A1").Value', 'Hello World1!'

EXEC @r = sp_OAGetProperty @WorkSheet, 'Cells', @Cells OUTPUT, 2, 2 -- Position de la cel
EXEC @r = sp_OASetProperty @Cells, 'Value', 'Hello World2!'
EXEC @r = sp_OASetProperty @Cells, 'Font.Bold', 1                   -- gras
EXEC @r = sp_OASetProperty @Cells, 'Font.Colorindex', 3             -- rouge pour la poli
EXEC @r = sp_OASetProperty @Cells, 'Interior.ColorIndex', 4         -- vert pour le fond
EXEC @r = sp_OASetProperty @Cells, 'Borders.ColorIndex', 5          -- bleu pour les bord

EXEC @r = sp_OASetProperty @Excel, 'ActiveWorkbook.Worksheets(1).Cells(3,3).Value', 'Hell
EXEC @r = sp_OASetProperty @Excel, 'ActiveWorkbook.Worksheets(1).Cells(3,3).Font.Colorind

EXEC sp_OAMethod @Excel, 'ActiveWorkbook.Save'
EXEC sp_OAMethod @Excel, 'Workbooks.Close'
EXEC sp_OAMethod @Excel, 'Close'

EXEC sp_OADestroy @Cells
EXEC sp_OADestroy @WorkSheet
EXEC sp_OADestroy @WorkBook
EXEC sp_OADestroy @WorkBooks
EXEC sp_OADestroy @Excel
```

Sauvegardes .bak et .trn

Tout d'abord, il faut distinguer l'archivage des bases (copie d'un .mdf en .bak) de celui des logs (.ldf en .trn) appelés aussi *journaux de transaction*.

Backup des bases

Il est recommandé d'effectuer automatiquement celui un backup des bases toutes les nuits, à l'aide d'un job (dans SSMS, en bas de l'arborescence, menu *Travaux*). Un deuxième pourra s'occuper de supprimer les .bak après une certaine durée de rétention qui peut dépendre de l'espace disponible sur le serveur.

Exemple de sauvegarde sur un disque dur du serveur[12] (et non pas un du client où SSMS est lancé) :

```
declare @chemin as varchar(255) = 'C:\' + CONVERT(char(10), GetDate(),126) + '-sugarcrm.b
BACKUP DATABASE sugarcrm
TO DISK = @chemin
   WITH FORMAT,
   MEDIANAME = '',
   NAME = 'Full Backup';
GO
```

Journaux

Il existe une fonction pour lire les journaux non archivés :

```
SELECT * FROM fn_dblog(NULL, NULL)
```

Afin de gagner de la place, ces logs doivent par contre être supprimés régulièrement (c'est sans incidence sur l'utilisation du système). Il existe trois façons de les tronquer :

1. `DBCC SHRINKFILE (N'MaBase_log' , 0, TRUNCATEONLY)`. *DBCC* est le sigle de *DataBase Console Commands* (commandes en console de base de données), c'est un ensemble de commandes sur les bases[13].
2. Clic droit sur la base, tâches, réduire... Fichiers, Type de fichier : Journal.
3. Clic droit sur la base, tâches, détacher... (la base disparait ensuite de la liste), déplacer le .ldf, puis clic droit sur le serveur, joindre... En sélectionnant le .mdf, un nouveau .ldf vierge sera automatiquement créé avec.

Pour le définir automatiquement, il faut créer un job : *Agent SQL server*, *Travaux*, Clic droit : *Nouveau travail*[14].

Restaurations

Pour restaurer une base de données dans SSMS, on peut faire un clic droit sur *Bases de données*, puis *Restaurer la base de données*.... On peut ensuite choisir d'écraser l'originale ou de renommer celle issue de l'archive.

Si la base à restaurer n'existe pas encore sur le serveur car elle provient d'un autre, la créer vide avec un clic droit sur *Bases de données*, *Nouvelle base de données*.

Il faut effectuer un clic droit sur la base à restaurer, puis *Restaurer la base de données* et sélectionner le .bak.

Sinon en SQL ça donne[15] :

```
RESTORE DATABASE MaBase
FROM DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\Backup\2016-0
WITH REPLACE
```

Références

1. http://www.mysql.fr/products/workbench/
2. http://www.thegeekstuff.com/2014/03/mssql-to-mysql-stored-procedure/

3. https://msdn.microsoft.com/fr-fr/library/ms190312.aspx
4. http://stackoverflow.com/questions/87735/how-do-you-transfer-or-export-sql-server-2005-data-to-excel
5. https://msdn.microsoft.com/fr-fr/library/ms141683.aspx
6. https://msdn.microsoft.com/fr-fr/library/ms141036.aspx
7. http://www.excel-sql-server.com/excel-import-to-sql-server-using-linked-servers.htm
8. http://www.excel-sql-server.com/excel-import-to-sql-server-using-distributed-queries.htm
9. https://msdn.microsoft.com/fr-fr/library/ms162802(v=sql.100).aspx
10. http://sqlindia.com/copy-move-files-folders-using-ole-automation-sql-server/
11. https://msdn.microsoft.com/fr-fr/library/office/ff840443.aspx
12. https://msdn.microsoft.com/fr-fr/library/ms186865%28v=sql.120%29.aspx?f=255&MSPPError=-2147217396
13. https://msdn.microsoft.com/fr-fr/library/ms188796(v=sql.120).aspx
14. http://communitybi.blogspot.fr/2011/12/comment-creer-un-job-dans-sql-server.html
15. https://msdn.microsoft.com/fr-fr/library/ms186858%28v=sql.100%29.aspx

Messages d'erreur

Un PRINT ne renvoie que des lignes blanches

C'est que la variable qu'il tente d'afficher a été concaténée avec `NULL`.

Un SELECT isnull(...) renvoie quand-même NULL

C'est qu'il contient une jointure sans aucune correspondance.

Les procédures stockées sont utilisables mais invisibles par l'explorateur d'objets du Management Studio

L'installation du module SQL_Search.exe[1] peut réparer cela. De plus cette extension propose une interface pour lancer des recherches de chaines de caractères dans la structure (pas dans les données).

Sinon elles sont accessibles en lecture seule avec `sp_helptext` :

```
-- Consultation
SELECT name
FROM sysobjects syso
order by name
-- Affichage
sp_helptext Procedure1
```

En PHP

Échec de l’ouverture de session / Login failed for usernam

Se produit quand on se connecte avec un compte SQL alors que c'est interdit. Pour les autoriser, dans SSMS, clic droit sur le serveur, Propriétés, Sécurité, cocher le deuxième bouton radio : *Authentification Windows et SQL Server*[2].

Erreur de contexte

Le mot de passe du compte a expiré.

Fatal error: Call to undefined function sqlsrv_connect()

Installer le pilote correspondant à la version de PHP du serveur Web :

1. Télécharger sur https://www.microsoft.com/en-us/download/details.aspx?id=20098.
2. Copier dans le dossier PHP (ex : C:\Program Files (x86)\EasyPHP\binaries\php\php_runningversion\ext).
3. Ajouter à PHP.ini.
4. Redémarrer le serveur Web.

L’autorisation SELECT a été refusée sur l’objet…

Cocher les rôles

This extension requires the Microsoft ODBC Driver 11 for SQL Server

Installer le pilote depuis https://www.microsoft.com/en-us/download/details.aspx?id=36434.

Unable to initialize module. Module compiled with module API=x. PHP compiled with module API=y. These options need to match

Se procurer une autre DLL à renseigner dans PHP.ini.

Messages système

Avertissement : la valeur NULL est éliminée par un agrégat ou par une autre opération SET

Résolu en remplaçant :

```
isnull(sum(
```

par

```
sum(isnull(
```

Ou encore :

```
where a = b
```

par

```
where isnull(a,0) = isnull(b,0)
```

Chargement en masse : DataFileType a été spécifié à tort comme étant char. DataFileType sera considéré comme étant widechar, parce que le fichier de données a une signature Unicode.

Voir ci-dessous.

Chargement en masse : DataFileType a été spécifié à tort comme étant widechar. DataFileType sera considéré comme étant char, parce que le fichier de données n'a pas de signature Unicode.

Se produit lorsque le paramètre "DATAFILETYPE" est renseigné pour importer de l'Unicode dans un "BULK INSERT" :

```
BULK INSERT MaTable1
FROM 'D:\MonFIchier1.csv'
WITH (
 FIELDTERMINATOR = ';',
 ROWTERMINATOR = '\n',
 DATAFILETYPE = 'widechar'
)
```

Comme SQL Server ne supporte pas UTF-8, il faut réencoder le fichier en UTF-16[3].

Échec de UPDATE car les options SET suivantes comportent des paramètres incorrects : 'ARITHABORT'.

Cela se produit quand on tente un UPDATE sur un champ protégé[4]. Peut-être utiliser un pilote ODBC.

Échec de la conversion de la date et/ou de l'heure à partir d'une chaîne de caractères.

Forcer la conversion en date :

```
INSERT INTO Table1 VALUES (convert(datetime, 'Date1', 121));
```

Si cela persiste, c'est que le résultat comporte une date impossible, comme le 31 février, juin, ou autres.

Erreur de conversion des données à charger en masse (troncation)

Lors d'une insertion, la valeur d'un champ dépasse ce qui était prévu. Par exemple en mode création il est en `varchar(32)` et qu'à la ligne indiquée par l'erreur on trouve une phrase de 33 lettres.

Échec de la conversion de la valeur varchar en type de données int.

Survient quand on compare un texte avec un nombre. Dans cet exemple il suffit de remplacer `2` par `'2'` :

```
select *
FROM Table1
where Champ1 = 2
```

Cela peut aussi se produire quand le premier résultat d'un `case` est un entier et le second une chaine :

```
select case when x=y then Entier1 else Chaine1 end
-- doit devenir
select case when x=y then convert(varchar,Entier1,112) else Chaine1 end
```

Erreur de conversion des données varchar en int.

S'il y a une comparaison avec un littéral, le mettre entre apostrophe.

Échec du chargement en masse. Valeur NULL attendue dans la ligne du fichier de données 1, colonne 1. La colonne de destination est définie comme NOT NULL.

Lors d'une insertion, la valeur d'un champ qui doit être non nulle n'est pas définie.

Erreur lors de la localisation de Server/Instance spécifié

Lors d'une jointure entre deux serveurs, vérifier que les ports 1433 TCP et UDP des serveurs sont ouverts.

Si oui, recréer le serveur lié en indiquant le login et mot de passe distant tout en bas du menu "Sécurité", dans "Seront effectuées dans ce contexte de sécurité".

Fatal error: Allowed memory size of 134217728 bytes exhausted

Augmenter la valeur de la ligne *memory_limit = 128M* du fichier php.ini. Puis relancer IIS ou Apache.

Il y a moins de colonnes dans l'instruction INSERT que de valeurs spécifiées dans la clause VALUES

Revoir la syntaxe de l'insertion[5], ex :

```
INSERT INTO Table1 VALUES ('Ligne1_Champ1', 'Ligne1_Champ2'), ('Ligne2_Champ1', 'Ligne2_C
```

Il manque un agrégat dans la liste de définition d'une instruction UPDATE

Un `UPDATE` n'arrive pas à sélectionner les champs à mettre à jour dans le `WHERE`. Essayer d'exécuter cette sous-requête individuellement.

Impossible d'appeler des méthodes sur char

Un champ doit avoir été tapé deux fois, ex : *Table1.Champ1.Champ1*.

Impossible d'extraire une ligne du fournisseur OLE DB "BULK" du serveur lié "(null)"

Lors d'une importation de fichier dans une table (`BULK INSERT`), le nombre de champs entre les deux ne correspond pas.

Impossible d'insérer une valeur explicite dans la colonne identité de la table *table* quand IDENTITY_INSERT est défini à OFF.

Au moins une clé de la table s'incrémente automatiquement, et ne peut pas être définie par une requête. Il faut soit la retirer des insertions, soit en désactiver la contrainte (ce qui ne stoppe pas l'auto-incrémentation) :

```
SET IDENTITY_INSERT table ON
```

Impossible d'utiliser le prédicat CONTAINS ou FREETEXT sur table ou vue indexée, car il n'y a pas d'index de texte intégral

Utilisez `sp_fulltext_database` pour activer la recherche en texte intégral dans la base[6].

Impossible de créer index sur la vue car celle-ci n'est pas liée au schéma

Il faut recréer la vue avec l'option `WITH SCHEMABINDING`.

Impossible de lier au schéma vue car le nom n'est pas valide pour la liaison au schéma.

Il ne faut pas nommer la base de données dans la commandes, ni oublier le mot "dbo" : `FROM [dbo].[table1]`.

L'autorisation SELECT a été refusée sur l'objet...

Dans *SSMS*, *Sécurité*, *Connexions*, il faut éditer le compte concerné dans *Rôles du serveur*, et cocher les cases manquantes.

L'identificateur en plusieurs parties "xxx" ne peut pas être lié

- Un champ de la sélection est absent des tables.
- Dans le cas d'une sélection de sélection, une table.champ de l'imbriquée n'est plus dans la seconde, il faut donc appeler le champ dans cette table en préfixe.
- Un `UPDATE` avec une jointure sans `FROM`.

L'index se trouve en dehors des limites du tableau

Cela peut se produire quand on manipule une grande base avec la version Express sur SSMS 2008.

Il faut alors passer par le code SQL plutôt que par SSMS, ou bien migrer en SSMS 2016.

La clause ORDER BY n'est pas valide dans les vues, les fonctions inline, les tables dérivées, les sous-requêtes et les expressions de table communes, sauf si TOP ou FOR XML est également spécifié

On peut utiliser *top* après *select* pour trier une sous-requête avec *order by*. Si le *top*

La colonne 'xxx' a été spécifiée plusieurs fois pour 'Z'

Lors d'un `SELECT * FROM (SELECT * FROM X join Y) Z`, les champs de X qui ont le même nom que ceux de Y font doublon dans Z. Il faut donc éviter d'utiliser *.

La colonne n'est pas valide dans la clause HAVING parce qu'elle n'est pas contenue dans une fonction d'agrégation ou dans la clause GROUP BY.

Placer simplement la colonne citée par l'erreur dans le `group by` au-dessus du `having`.

La conversion d'un type de données varchar en type de données (small)datetime a créé une valeur hors limites

Forcer la conversion en date :

```
INSERT INTO Table1 VALUES (convert(datetime, 'Date1', 121));
```

Si cela persiste, c'est que le résultat comporte une date impossible, comme le 31 février, juin, ou autres.

La définition de l'objet 'ProcédureStockée1' a changé depuis la compilation

La procédure stockée a été mise à jour pendant son exécution. Il faut donc la relancer.

La sous-requête a retourné plusieurs valeurs. Cela n'est pas autorisé quand la sous-requête suit =, !=, <, <= , >, >= ou quand elle est utilisée en tant qu'expression.

Un champ dans une requête imbriquée envoie plusieurs résultat au lieu d'un seul. Il est inclut dans l'opérande d'une formule avec opérateur de comparaison. Utiliser `TOP 1`.

Parfois un trigger doit être désactivé pour régler cela.

Le fournisseur OLE DB "Microsoft.ACE.OLEDB.12.0" n'a pas été enregistré.

Il faut l'installer depuis : https://www.microsoft.com/fr-FR/download/details.aspx?id=23734.

Le fournisseur OLE DB "Microsoft.Jet.OLEDB.4.0" du serveur lié "(null)" a retourné le message "Erreur non spécifiée".

Commencer par redémarrer le service SQL.


Cette section est vide, pas assez détaillée ou incomplète.

Le fournisseur OLE DB 'Microsoft.Jet.OLEDB.4.0' ne peut pas être utilisé pour les requêtes distribuées, car le fournisseur est configuré pour s'exécuter en mode STA.


Cette section est vide, pas assez détaillée ou incomplète.

Le [*sic*] identificateur qui commence par '...' est trop long. La longueur maximale est 128.

Probablement dû à l'utilisation des guillemets comme séparateur de chaine. Remplacer :

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

par

```
SET ANSI_NULLS OFF
GO
SET QUOTED_IDENTIFIER OFF
GO
```

Le jeu de sauvegarde contient la sauvegarde d'une base de données qui n'est pas la base de données

Utiliser :

```
WITH REPLACE
```

A la fin de la commande de restauration.

Le nom ou le numéro de colonne des valeurs fournies ne correspond pas à la définition de la table.

Le nombre de champ à insérer n'est pas le même que celui de la table, il faut donc soit les nommer entre

parenthèses après la table, soit ajouter des valeurs par défaut aux champs manquants :

```
-- Si la table a trois champs :
INSERT INTO table1 (champ1, champ2) VALUES (champ1, champ2)
-- ou
INSERT INTO table1 VALUES (champ1, champ2, 0)
```

Le nombre d'expressions de valeurs de ligne de l'instruction INSERT dépasse le nombre maximal autorisé de 1000 valeurs de ligne

Scinder en plusieurs requêtes.

Le type de données de l'opérande datetime n'est pas valide pour l'opérateur sum

Operand data type datetime is invalid for sum operator.

Il convient de préférer *DATEADD* à *SUM*.

Les colonnes de la table ne correspondent pas à une clé primaire existante ou à une contrainte UNIQUE

Il faut définir une contrainte d'unicité[7].

Les données de chaîne ou binaires seront tronquées

L'insertion d'un enregistrement contient une valeur qui ne rentre pas dans un champ varchar(x). Il faut donc la tronquer ou bien modifier la taille du champ (ex : `left(v, 50)`).

Sinon il s'agit d'un caractère indésirable qui s'est glissé dans une chaine à insérer, le retrouver par exemple par dichotomie.

Les requêtes hétérogènes requièrent les options ANSI_NULLS et ANSI_WARNINGS pour être définies pour la connexion. Cela assure la cohérence sémantique de la requête. Activez ces options et relancez la requête.

Placer les commandes suivantes avant de recréer la procédure stockée :

```
SET ANSI_NULLS ON
SET ANSI_WARNINGS ON
```

Sinon, vérifier qu'elles ne sont pas définies à *OFF* à un autre endroit.

Sinon, cela peut fonctionner depuis une requête séparée préalable.

Les valeurs DEFAULT et NULL ne sont pas autorisées comme valeurs d'identité explicites.

Provoqué par un `INSERT` d'une valeur nulle dans un champ clé `AUTO_INCREMENT`. Il suffit donc de ne pas modifier ce champ du tout quand on modifie son enregistrement.

Nom de colonne non valide : 'xxx'

Remplacer les guillemets par des apostrophes : "xxx" → 'xxx'.

Ouvrez les guillemets après la chaine de caractères

Le nombre d'apostrophes délimitant les chaines de caractères est impair, la balance n'est donc pas à l'équilibre. Sinon un " est utilisé comme un ' à tort.

Procédure stockée 'xxx' introuvable.

Voir ci-dessous.

SQL Server a bloqué l'accès à procédure 'sys.sp_OACreate' du composant 'Ole Automation Procedures', car ce composant est désactivé dans le cadre de la configuration de la sécurité du serveur

Il faut activer le module comme expliqué dans le chapitre Microsoft SQL Server/Importer_et_exporter#sp_OACreate.

Syntaxe incorrecte vers 'xxx'.

Lors de la sauvegarde d'une procédure stockée, si le curseur contient une sélection son contenu tente d'être exécuté indépendamment du reste du code. Il faut donc faire un clic ailleurs.

Syntaxe incorrecte vers '+'.

Les variables et les `case` sont interdits dans les `from`.

Syntaxe incorrecte vers le mot clé 'case'.

Les variables et les `case` sont interdits dans les `from`.

Syntaxe incorrecte vers le mot clé 'end'.

Si en passant la souris sur le `end` il est écrit "Attendu CONVERSATION", c'est qu'il suit un `if` vide.

Syntaxe incorrecte vers le mot clé 'group'

- Lors de plusieurs SELECT imbriqués, il faut nommer la sélection. L'exemple suivant fait la somme des heures de chaque personne, dont une qui a deux noms :

```
SELECT SUM(Heures), Personnes FROM (
  SELECT SUM(Heures), case Personnes when 'Mr X²' then 'Mr X' else Personnes end as Perso
  FROM TableH
  group by Personnes) h -- Erreur sans cette lettre
group by Personnes
order by Personnes
```

Un agrégat ne peut pas apparaître dans une clause WHERE à moins que ce ne soit une sous-requête contenue dans une clause HAVING ou une liste de sélection, et que la colonne à agréger soit une référence externe.

Il y a un `count()` sur un champ de la requête dans sa sous-requête.

Une erreur de dépassement arithmétique s'est produite lors de la conversion de varchar en type de données numeric.

Si l'augmentation de la taille de la variable de destination ne suffit pas (ex : `decimal(38,10)`), il faut lever les exceptions, par exemple en ajoutant une condition en amont (ex : `where (isnumeric(Chaine) = 1 and...)`), ou `TRY CATCH`[8], voire `THROW` dans SQL 2014[9].

Une seule expression peut être spécifiée dans la liste de sélection quand la sous-requête n'est pas introduite par EXISTS.

Cela arrive quand dans la close WHERE une condition IN renvoie vers plusieurs champs au lieu d'un seul (ex : `where Num_Client in (select * from...)` au lieu de `where Num_Client in (select Numero_Client from...)`).

Une valeur explicite de la colonne identité de la table ne peut être spécifiée que si la liste des colonnes est utilisée et si IDENTITY_INSERT est défini sur ON.

Provoqué par un `INSERT` d'une valeur dans un champ clé `AUTO_INCREMENT`. Il suffit donc de ne pas modifier ce

champ du tout quand on modifie son enregistrement.

Violation de la contrainte UNIQUE KEY

Lors d'une insertion dans une table dont la clé ne s'incrémente pas automatiquement, il faut soit y remédier soit la spécifier pour chaque enregistrement de l'insertion.

Références

1. https://www.red-gate.com/products/sql-development/sql-search/
2. http://blog.sqlauthority.com/2014/01/17/sql-server-fix-login-failed-for-user-username-the-user-is-not-associated-with-a-trusted-sql-server-connection-microsoft-sql-server-error-18452/
3. https://aliiraza.wordpress.com/2012/06/05/bulk-load-datafiletype-was-incorrectly-specified-as-widechar-datafiletype-will-be-assumed-to-be-char-because-the-data-file-does-not-have-a-unicode-signature/?preview=true&preview_id=103&preview_nonce=3350b4ba67
4. http://technet.microsoft.com/fr-fr/library/ms189118%28v=sql.105%29.aspx
5. http://technet.microsoft.com/fr-fr/library/dd776381%28v=sql.105%29.aspx
6. http://doc.nuxeo.com/plugins/viewsource/viewpagesrc.action?pageId=4685883
7. http://office.microsoft.com/fr-fr/help/les-colonnes-de-la-table-0s-ne-correspondent-pas-a-une-cle-primaire-existante-ou-a-une-contrainte-unique-HP003083867.aspx
8. https://msdn.microsoft.com/en-us/library/ms175976.aspx
9. https://msdn.microsoft.com/fr-fr/library/Ee677615%28v=SQL.120%29.aspx

- http://technet.microsoft.com/fr-fr/library/cc645611%28v=sql.105%29.aspx

SugarCRM

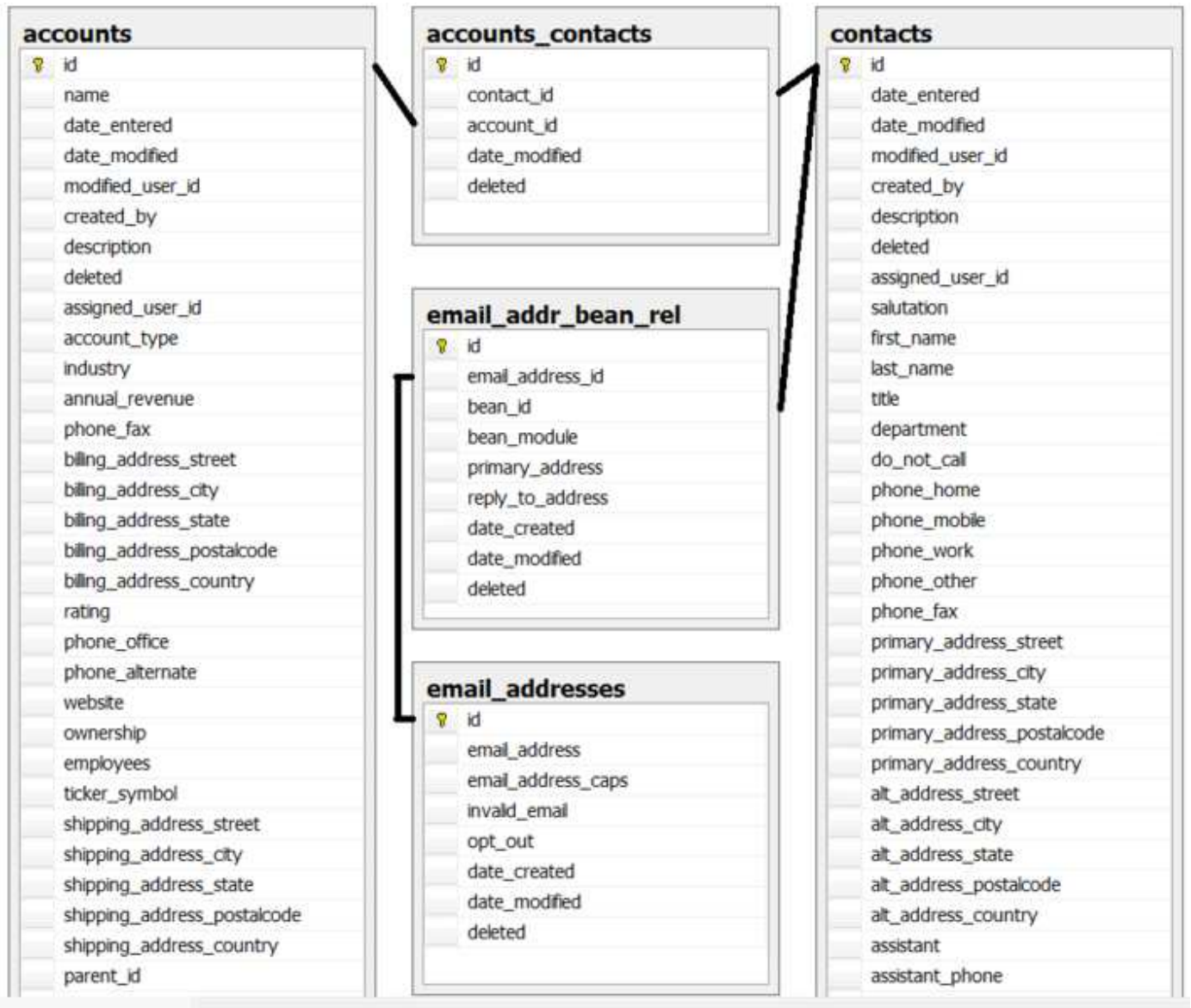
SugarCRM est logiciel de gestion de la relation client compétitif et open source, sous forme de site PHP qui être configuré pour MSSQL (ou MySQL).

Installation

Il existe des versions gratuites payantes du logiciel[1], ainsi que des modules complémentaires également gratuits et payants[2]. La présente page traite de la version gratuite à télécharger sur https://sourceforge.net /projects/sugarcrm/files/latest/download?source=files.

Une fois décompressée et placée dans un répertoire de serveur HTTP (ex : Apache ou IIS), il suffit d'y accéder dans un navigateur par le nom du dossier (ex : http://localhost/SugarCRM), et d'y renseigner le nom de la base de données (ex : SugarCRM) et le mot de passe associé, précédemment défini dans Microsoft SQL Server Management Studio (nouvelle connexion).

Architecture de la base



La base est à la première forme normale, et certaines tables font juste le lien entre les clés primaires d'autres :

- *accounts_contacts* : associe un contact à une entreprise, avec la date d'association.
- *accounts_opportunities* : associe une entreprise à un devis, avec date de mise à jour.
- *email_addr_bean_rel* : associe une adresse email à une personne. En effet, bien qu'un même individu puisse avoir une fiche employé (*users*), une contact (*contacts*) et une prospect (*leads*) séparées, son adresse email est stockée à part et est la même pour tous ses rôles.

Requêtes

Insertion de comptes et de contacts liés :

```
insert into accounts (id, name)
values ('1', 'Entreprise1'),
values ('2', 'Entreprise2')

insert into contacts (id, last_name, first_name)
values ('1', 'Doe', 'Jane'),
values ('2', 'Doe', 'John')

insert into accounts_contacts(id, contact_id, account_id, date_modified)
values ('1', '1', '1', convert(datetime,getdate(),121))  -- Met Jane Doe dans l'entrepris
values ('2', '2', '2', convert(datetime,getdate(),121))
```

Liste des entreprises avec leurs contacts :

```
select *
from accounts a
inner join accounts_contacts ac on ac.account_id = a.id
inner join contacts c on c.id = ac.contact_id
```

Après avoir ajouté des adresses emails, liste des contacts avec leurs emails :

```
select c.first_name + ' ' + c.last_name, e.email_address
from contacts c
inner join email_addr_bean_rel er on er.bean_id = c.id
inner join email_addresses e on e.id = er.email_address_id
```

Références

1. http://www.sugarcrm.com/fr/mic-editions
2. https://github.com/sugarcrm

Voir aussi

Liste d'autres logiciels compatibles MSSQL :

Gratuits

- Apache OFBiz
- phpBB
- WordPress

Payants

- Atheneo
- Sage