

Etablissement Inter – Etats d'Enseignement Supérieur Représentation du Cameroun

CENTRE D'EXCELLENCE TECHNOLOGIQUE PAUL BIYA BP 13 719 Yaoundé (Cameroun) Tél. (237) 242 72 99 57/ 242 72 99 58

Site web: www.iaicameroun.com E-mail: info@iaicameroun.com

<u>UE</u>: ADMINISTRATION DES BASES DE DONNEES SOUS SQL-SERVER

COURS PROPOSE ET DISPENSE PAR : Mme Belinga Estelle Master En Sécurité Des Systèmes Numériques

Année Académique :2023-2024

Plan Du Cours

<u>Ière Partie</u>: Présentation de SQL Server

<u>IIe Partie</u>: Prise en main de SQL Server

IIIe Partie: Travaux Pratiques

Ière Partie: Présentation de SQL SERVER

Introduction

Microsoft SQL Server est un SGBDR développé et commercialisé par Microsoft, c'est un logiciel permettant le stockage de données de façon cohérente et organisée de manière à pouvoir

facilement consulter et modifier leur contenu. Il utilise le langage T-SQL (Transact-Sql) pour ses

requêtes, c'est une implémentation du SQL qui prend en charge les procédures stockées ainsi que

les déclencheurs. Il est originellement multi base et multishema Par exemple la requête suivante

lie deux tables de deux bases de données différentes :

SELECT *

FROM BASE A.dbo.TABLE1 AS T1 INNER JOIN BASE B.dbo.TABLE2 AS T2

ON T1.ID = T2.ID;

SQL SERVER regorge de plusieurs éditions, qui offrent différents niveaux de services. Les deux

versions gratuites sont : l'édition Developer comprenant toutes les fonctionnalités pour etre

utilisées comme base de données de développement et de test dans des environnements non dédiés

à la production, ainsi qu'une version Express idéale pour le développement et la production

d'applications de bureau, d'applications web et de petites applications serveur. Les autres éditions

ne sont pas gratuites. Microsoft vend une version Entreprise qui offre une sélection complète de

fonctionnalités de SQL Server, ainsi qu'une édition Standard avec des fonctionnalités partielles et

d'autres limites telles que la taille de la mémoire.

Devoir : Présenter les caractéristiques de chaque édition.

C'est grâce au protocole TDS (Tabular Data Stream), qu'un client peut se connecter à SQL

SERVER (protocole utilisé pour le transfert de requêtes et de réponses entre les clients et SQL

SERVER).

La 1ère version de SQL SERVER est sortie en 1989. Aujourd'hui il est à la version 22.

Le fonctionnement I.

On distingue deux types de serveurs sur lesquels l'environnement SQL Server peut être déployé.

Le serveur OLTP (Online Transactionnal Processing) qui est un serveur transactionnel,
 i.e. un serveur sur lequel un certain nombre de transactions de type peuvent être exécutées.
 La plupart du temps les systèmes OLTP sont utilisés pour la saisie des commandes, les transactions financières, la gestion de la relation client et les ventes etc. ces systèmes comptent un grand nombre d'utilisateurs qui effectuent des transactions courtes et retournent relativement peu d'enregistrements.

En résumé, un serveur OLTP possède les caractéristiques suivantes :

- Les transactions impliquent de petites quantités de données
- Possède un grand nombre d'utilisateurs
- Les temps de réponses sont assez rapides Demande une grande disponibilité
- Le serveur OLAP (Online Analytical Processing) est un serveur décisionnel. Il permet de consolider les données venant de sources multiples grâce à un ETL (extraction transformation loading qui permet d'extraire les données venant de sources multiples : Excel, XML, MySQL ... les transforme en un même format puis les charge dans un datawarehouse).

OLAP effectue une analyse multidimensionnelle des données d'entreprise et offre la possibilité d'effectuer des calculs complexes .Il est utilisé en entreprise pour la gestion de la performance, la planification, les prévisions, l'analyse, la budgétisation, les rapports financiers....Il fournit aux analystes et managers les informations dont ils ont besoin pour prendre des décisions. Néanmoins il ne comporte pas un grand nombre d'utilisateurs

En résumé un serveur OLAP permet :

- Une analyse rapide de données
- D'organiser et de comparer les données
- Effectuer du reporting
- D'effectuer des calculs complexes
- Manipuler un nombre important de données (pouvant aller à plusieurs Téra octets = 10^e12 octets)

II. Les composants de SQL Server

Microsoft SQL Server est composé de plusieurs logiciels qui s'exécutent sous forme de service (type d'applications qui s'exécutent en arrière-plan). SQL Server est composé de 05 principaux services :

- SQL SERVER: C'est le service moteur de base données et correspond à une instance SQL
 Server Ce composant s'exécute en tant que service Windows et est référencé sous le nom
 MSSQLSERVER pour l'instance par défaut et MSSQLSERVER\$nomInstance pour une instance nommée. Le nom d'une instance est défini lors de l'installation.
- SQL Server Integration Service (SSIS): C'est le composant d'Import/Export, de transfert et de transformation de données. Il intègre des assistants pour la création d'ETL (Extraction Transformation Loading).
- SQL Server Analysis Service (SSAS) : C'est le composant idéal pour les projets décisionnels. Ce composant est un outil d'analyse OLAP et de Data Mining de Microsoft et permet de construire des cubes OLAP (base de données à plusieurs dimensions : rolap, molap, holap avec la particularité selon laquelle molap est plus rapide et plus performante que rolap mais limité en taille (giga octets)).
- SQL Server Reporting Service (SSRS) : C'est grâce à ce composant que nous pouvons restituer nos données provenant de notre entrepôt de données sous forme de rapports (tableau, graphique etc....).
- Agent SQL: En charge de la surveillance de SQL Server, ce composant gère également l'exécution des tâches planifiées et le suivi des alertes. Il est directement lié à une instance SQL Server. Il est référencé par défaut dans le gestionnaire de service Windows sous le nom SQL Server Agent et par SQL Server Agent (nom instance) dans le cas d'une instance nommée.

III- Les outils

Voici quelques outils que l'on peut retrouver dans SQL Server:

- **SQL Server Management Studio (SSMS)**: il permet de réaliser toutes les opérations au niveau du moteur de BD.
- Le gestionnaire de configuration SQL Server : il permet de gérer tous les services liés à SQL Server, c'est-à-dire vous permet d'arrêter, démarrer ou d'interrompre divers services SQL SERVER

- SQL Server Profiler : il permet de suivre et d'analyser la charge de travail d'une instance SQL Server (déprécié)
- Azure Data Studio : anciennement appelé SQL Operations Studio permet de créer les bases de données, les interroger, effectuer des sauvegardes, restaurer des bases de données
- Le SQL CMD: il permet d'exécuter des requêtes, des scripts en ligne de commande.
- Le SQL SERVER DISTRIBUTED REPLAY: Vous aide à évaluer l'impact de futures mises à niveau de SQL SERVER. Il est également utilisé pour évaluer l'impact des mises à niveau du matériel et du système d'exploitation.
- Les évènements étendus (XEvents) : Ils permettent aux utilisateurs de collecter des données nécessaires pour surveiller, identifier ou résoudre les problèmes de performance.

IV- LES BASES DE DONNEES SQL SERVER

SQL Server héberge deux types de bases de données : les bases de données système et les bases de données utilisateur.

Les bases de données système contiennent des informations nécessaires au bon fonctionnement du moteur SQLServer et des services associés tandis que les bases de données utilisateur sont celles créées et utilisées par les utilisateurs. Parmi les bases de données système, on retrouve :

- Master : Elle enregistre toutes les informations relatives à SQL SERVER, en effectuant le suivi d'informations telles que les comptes d'utilisateur, les paramètres de configuration du système et les messages d'erreur du système. La base de données master enregistre également l'existence de toutes les bases de données, l'emplacement de leurs fichiers et permet l'initialisation de SQL SERVER. Du coup, SQL SERVER ne peut pas démarrer si la base de données master n'est pas disponible.
- Model: Offre un modèle ou prototype pour les nouvelles bases de données utilisateur. Lorsqu'une instruction Create Database est émise, le système crée la première partie de la bd en copiant le contenu de la bd model. Le reste est ensuite rempli de pages vides. Si vous modifiez la bd model, toutes les bases de données crées ultérieurement héritent des modifications apportées. (exemple création des tables, procédures stockées...).
- Msdb: Elle est utilisée par l'Agent SQL pour planifier les alertes et les travaux (sauvegarde, restauration...).

- Resource : C'est une base de données en lecture seule qui contient tous les objets de SQL SERVER. La bd resource ne contient ni données utilisateurs, ni métadonnées utilisateur (différence entre les deux). Cette bd doit être modifiée par un spécialiste du support technique Microsoft.
- Tempdb : Offre une zone de stockage pour les tables temporaires et les autres besoins de stockage temporaire.

V- TACHES D'UN ADMINISTRATEUR DE BASE DE DONNEES SQL SERVER :

Un administrateur de base de données SQL SERVER doit être capable de :

- Installer, configurer le serveur et administrer les bases de données
- Gérer l'espace de stockage physique
- Veiller à l'intégrité des données stockées dans les bases
- Mettre en place des plans de sauvegarde, de maintenance
- Mettre en place des solutions de maintien de la continuité de la production
- Veiller au maintien de la performance de l'accès aux données
- Participer à la veille technologique (toujours se tenir informé des innovations sur le secteur)

VI- Structure d'une BD SQL Server

Une BD SQL Server est composée de deux types de fichiers, à savoir les fichiers de données et les fichiers journaux.

Les fichiers de données : il en existe deux types : les fichiers primaires de données (obligatoires et d'extension .mdf) qui renferment le catalogue de la base de données , chaque bd comprend un fichier de données primaire ; les fichiers secondaires de données (facultatifs et d'extension .ndf) qui renferment les objets et les données utilisateurs.

Les fichiers journaux : avec pour extension .ldf, ils enregistrent toutes les modifications de la BD. Ils contiennent les informations qui seront utilisées pour la restauration de la bd. Chaque bd doit posséder au moins un fichier journal.

IIe Partie: PRISE EN MAIN DE SQL SERVER

Lab:

Installer SQL SERVER 2017 DEVELOPPER

1- Se connecter à une instance de SQL Server

- Démarrez SQL Server Management Studio
- Au niveau de la rubrique **Type de serveurs**, sélectionnez **Moteur de base de données**.
- Pour le nom du serveur, entrez le nom de votre instance SQL
- Au niveau de la rubrique Authentification, sélectionnez Authentification Windows.
- Cliquez sur Se connecter.

2- Création d'une base de données.

Pour créer une base de données, on peut soit utiliser SQL Server Management Studio, soit utiliser le Transact SQL.

a. Utilisation de SQL Server Management Studio

- Dans l'explorateur d'objets, connectez-vous à une instance du moteur de données SQL
 Server.
- Cliquez avec le bouton droit sur **Base de données**, puis cliquez sur **Nouvelle base de données**.
- Dans **Nouvelle base de données**, entrez le nom de la base de données.
- Pour créer la base de données, en acceptant toutes les valeurs par défaut, cliquez sur **OK**.
- Pour modifier les valeurs par défaut des fichiers primaires et des fichiers journaux ou de transaction, cela se fait dans la grille Fichier de la base de données, cliquez sur la cellule appropriée puis entrez la nouvelle valeur.
- Pour modifier les options de la base de données, sélectionnez l'onglet **Options**, puis apporter les modifications de votre choix.

- Pour ajouter une propriété étendue (propriétés personnalisées) à la base de données, sélectionnez **Propriété étendue**.
 - Dans la colonne Nom, entrez le nom de la propriété étendue ;
 - Dans la colonne Valeur, entrez le texte de la propriété étendue ; ensuite cliquez sur OK

b. Utilisation de Transact SQL

Dans la barre d'outils standard, cliquez sur Nouvelle requête :

```
USE master;
GO
CREATE DATABASE GL
ON
(NAME=GL \ dat,
FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\GL dat.mdf',
SIZE=10.
MAXSIZE = 50.
FILEGROWTH=5)
LOG ON
(NAME=GL log,
FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\GL\log.ldf',
SIZE=5, // en MO
MAXSIZE=25.
FILEGROWTH=5)
GO
```

3- Supprimer une base de données

a. Utilisation de SSMS (SQL Server Management Studio)

- Dans l'explorateur d'objets, connectez-vous à une instance d'objets du moteur de base de données.
- Développez le dossier **Base de données**, cliquez avec le bouton droit sur la base de données à supprimer puis cliquez sur **Supprimer**.
- Vérifiez que la base de données correcte est sélectionnée puis cliquez sur **OK**.

- ; // qte d'espace ajouté au fichier à chaque fois qu'un nouvel espace est requis .il doit être inférieur ou égal à maxsize

b. Utilisation de Transact SQL

```
USE master;
GO
DROP DATABASE GL;
GO
```

4- Renommer une base de données

a. Avec SSMS

- Toujours vérifier qu'il n'y a aucune connexion ouverte à la base de données, il faut donc définir la base de données en mode mono-utilisateur, pour fermer toutes les connexions ouvertes et empêcher les autres utilisateurs de se connecter, pendant que le nom de la BD est modifié. Pour ce faire :
 - Cliquez avec le bouton droit sur la BD à modifier, puis cliquez sur **Propriétés**.
 - Dans la boite de dialogue Propriétés de la base de données, cliquez sur Options.
 - Dans l'option Restreindre l'accès, sélectionnez Utilisateur unique.
 - Si d'autres utilisateurs sont connectés à la base de données, un message Ouvrir les connexions apparait pour appliquer la propriété et fermer toutes les autres connexions; cliquez sur OUI.
- Dans l'explorateur d'objets, développez le nœud **Base de données**, cliquez avec le bouton droit sur la BD à renommer, puis cliquez sur **Renommez**.
- Entrez le nouveau nom de la base de données, puis cliquez sur **OK**.

b. Avec T-SQL

La syntaxe est :

```
USE master;
GO
ALTER DATABASE GL SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO
ALTER DATABASE GL MODIFY NAME=GLCOPY;
GO
ALTER DATABASE GLCOPY SET MULTI USER;
```

GO

5. Ajouter des fichiers de données ou de journaux

```
Use Master;
ALTER DATABASE GLCOPY
ADD FILE
 NAME = Glcopy2,
  FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Glcopy2.ndf,
  SIZE = 5MB,
  MAXSIZE = 100MB,
 FILEGROWTH = 5MB
);
GO
                        6. Ajouter la taille d'un fichier
USE master;
GO
ALTER DATABASE GLcopy
MODIFY FILE
 (NAME = gldat,
 SIZE = 20MB);
GO
                       7. Modifier le nom d'un fichier
USE master;
GO
ALTER DATABASE GLcopy
MODIFY FILE
 (NAME = gldat,
 NEWNAME = gldatcopy);
GO
```

8. Supprimer les fichiers de données ou de journaux

a. Avec SSMS

- Développer le dossier **Base de données**, cliquez avec le bouton droit sur la BD sur laquelle vous souhaitez supprimer le fichier, puis cliquez sur **Propriétés**.
- Sélectionnez Fichier.

- Dans la grille **Fichier de la base de données**, sélectionnez le fichier à supprimer, puis cliquez sur **OK**.

b. Avec Transact SQL

```
USE master;
GO
ALTER DATABASE GLCOPY REMOVE FILE GL_dat;
GO
```

La notion de schéma

Une base de données contient plusieurs objets tels que : les tables, les vues, les procédures stockées, les déclencheurs etc. Un schéma est une collection logique d'objets de base de données, son objectif est de rendre l'éxécution des requetes plus rapides et de faciliter la gestion de la sécurité d'accès des utilisateurs vers les objets, car il définit un ensemble de règles qui régissent une base de données ou un ensemble d'objets appartenant à des utilisateurs.

Le schéma par défaut sur SQL SERVER est dbo (database owner : proprietaire de base de données).

- Consulter la liste des schemas :
 - Select * from sys.schemas;
- Création d'un schéma
 - Create schema Glcopy authorization nom user;
- Modifier un schema
 - Alter schema Glcopy Transfer OBJECT::dbo.Etudiant / TRANSFER dbo.Etudiant
- Supprimer un schema:
 - Drop schema nom schema (il faut d'abord supprimer les objets contenus dans le schema)

9. Création d'une table

a. Avec SSMS

- Dans l'explorateur d'objets, développez le nœud **Base de données**, puis sélectionnez la base de données qui va contenir la nouvelle table.

- Cliquez avec le bouton droit sur le nœud **Table** de votre BD. Puis cliquez sur **Nouvelle table**.
- Tapez les noms des colonnes, choisissez le type de données, et spécifiez si les valeurs nulles sont autorisées pour chaque colonne.
- Pour spécifier d'avantages de propriétés pour une colonne, cliquez sur la colonne et choisissez les propriétés appropriées dans l'onglet des propriétés de la colonne.
- Pour spécifier une colonne comme clé primaire, cliquez avec le bouton droit sur la colonne et sélectionnez **Définir la clé primaire**.
- Pour créer les relations de clé étrangère, des contraintes de validation ou des index,
 cliquez avec le bouton droit dans le volet Concepteur de table et sélectionnez un objet dans la liste.
- Par défaut, la table est contenue dans le schéma DBO. Pour spécifier un schéma différent pour la table, cliquez avec le bouton droit dans le volet Concepteur de table, et sélectionnez Propriétés. Dans la liste déroulante Schéma, sélectionnez le schéma approprié.
- Dans le menu Fichier, choisissez Enregistrer nom de la table.
- Dans la boite de dialogue, choisissez un nom, pour la table et cliquez sur **OK**.

b. Avec Transact SQL

La syntaxe est:

```
USE GL;
GO
CREATE TABLE dbo. Etudiants
etudiantID INT NOT NULL PRIMARY KEY IDENTITY,
coursID INT NOT NULL,
nom NVARCHAR (50) NOT NULL,
prenom NVARCHAR (50),
pseudo NVARCHAR (50) NOT NULL UNIQUE,
FOREIGN KEY(coursID) REFERENCES Cours(coursID)
);
GO
```

c. Quelques types de données sous SQL Server

Un type de données est un attribut qui spécifie le contenu d'un objet.

- Les valeurs numériques exactes: BIGINT (8 octets), NUMERIC (p,s), BIG, SMALLINT (2 octets), DECIMAL (p,s), SMALLMONEY (4 octets), INT (4 octets), TINYINT (1 octet), MONEY (8 octets), bit;
- Les valeurs numériques approximatives : FLOAT, REAL ;
- Date et heure: DATE, TIME, DATETIME2 (année, mois, jour, heure, minute, seconde, fraction de seconde), DATETIMEOFFSET (année, mois, jour, heure, seconde, fraction de seconde, fuseau horaire), DATETIME (année, mois, jour, heures, minutes, secondes), SMALLDATETIME (les secondes sont nulles);
- Chaînes de caractères : CHAR, VARCHAR, TEXT ;
- Les chaînes de caractères Unicode (représente tous les caractères spécifiques aux différentes langues): NCHAR, NVARCHAR, NTEXT;
- Les chaînes binaires : IMAGE, VARBINARY (fichiers, image.), BINARY.
- Les autres types de données : XML, curseurs, types spatiaux-geometriques , types spatiaux-geographiques...

10. Supprimer une table

<u>NB</u>:

- Vous ne pouvez pas supprimer une table qui est référencée par une contrainte FOREIGN
 KEY, vous devez au préalable supprimer la contrainte ou la table qui la référence.
- Lorsqu'une table est supprimée, les règles et les valeurs par défaut sont supprimées et toutes les contraintes qui lui sont associées sont automatiquement supprimées.
- La suppression d'une table entraine automatiquement celle de toutes les relations qu'elle entretient

a. Sous SSMS

- Dans l'explorateur d'objets, développez le nœud **Base de données**, puis sélectionnez la base de données puis le nœud **Table**;
- Cliquez avec le bouton droit sur la table, puis dans le menu contextuel, cliquez sur **Supprimer**.

- Un message vous demande de confirmer la suppression, cliquez sur **OK**.

b. Sous Transact SQL

```
La syntaxe est:
```

USE GLCOPY;

GO

DROP TABLE dbo. Etudiants;

Voir supprimer une contrainte : alter table dbo.etudiant drop constraint nom_contrainte

11. Renommer une table

a. Sous SSMS

- Dans l'explorateur d'objets, cliquer avec le bouton droit sur la table sur laquelle vous souhaitez renommer et choisissez **Renommer**.
- Entrez un nouveau nom à la table puis validez.

b. Sous Transact SQL

La syntaxe est:

USE GLCOPY:

GO

EXEC sp_rename 'dbo. Etudiants', 'nouveauNom'; // pas besoin de mettre dbo sur le nouveau sinon il va le faire 2 fois

GO

12. Ajouter des colonnes à une table

a. Sous SSMS

- Dans l'explorateur d'objets, développez le nœud **Base de données**, puis sélectionnez la base de données puis Cliquez sur la table sur laquelle vous souhaitez ajouter des colonnes ;
- Faire un clic droit sur la rubrique Colonne, puis cliquez sur Nouvelle colonne.
- Entrez les nouvelles colonnes avec leur type puis valider.

b. Sous Transact SQL

La syntaxe est:

```
USE GL;

GO

ALTER TABLE dbo.Etudiants ADD

Nom_colonne1 type1,

Nom_colonne2 type2,

...

Nom_colonneN typeN;

GO
```

13. Supprimer une colonne

a. Sous SSMS

- VII- Rechercher la table que vous souhaitez supprimer les colonnes, développez-la pour exposer les noms des différentes colonnes ;
- VIII- Cliquez avec le bouton droit sur la colonne à supprimer, puis choisissez **Supprimer**.
- IX- Dans la boîte de dialogue Supprimer un objet, cliquez sur OK;
 Si la colonne contient des contraintes ou d'autres dépendances, un message d'erreur s'affichera dans la boite de dialogue Supprimer un objet. Résolvez l'erreur en supprimant les contraintes référencées.

<u>NB</u>:

- X- Vous ne pouvez pas supprimer une colonne ayant une contrainte CHECK. Vous devez d'abord supprimer la contrainte
- XI- Vous ne pouvez pas supprimer une colonne comportant des contraintes PRIMARY KEY ou FOREIGN KEY ou d'autres dépendances sauf si vous utilisez le concepteur de tables, dans le cas contraire vous devez d'abord supprimer toutes les dépendances de la colonne.

b. Sous Transact SQL

La syntaxe:

USE GLCOPY;

GO

ALTER TABLE dbo. Etudiants

DROP COLUMN nom;

//modifier le type d'une colonne alter table dbo.etudiant alter column nom varchar(50);

14. Renommer une colonne //à vérifier

a. Sous SSMS

- XII- Dans l'explorateur d'objets, cliquer avec le bouton droit sur la colonne sur laquelle vous souhaitez renommer et choisissez **Renommer**.
- XIII- Entrez un nouveau nom à la colonne puis validez.

b. Sous Transact SQL

USE GL;

GO

EXEC SP_RENAME 'dbo.Etudiants.Nom', 'nouveau_nom_de_la_colonne', 'column';

GO

15. Insérer des lignes dans la table

INSERT INTO dbo. Etudiants (etudiantID, nom, p renom, filière, niveau)

VALUES (1, N'NGNOULAYE', N'Jeff', N'Génie Logiciel', 3);

16.Interroger la table et afficher les résultats

SELECT * FROM dbo. Etudiants , update, like, between, jointures, group by, order by, having, top (5)*, date de naissance etc, étudiants dont la plus grande note est sup au double de la moyenne

SEANCE DE TRAVAUX PRATIQUES + CC ou Thèmes d'exposés

17.Les procédures stockées

Une procédure stockée dans SQL Server est un groupe d'une ou plusieurs instructions T-SQL dans le but d'être réutilisées facilement (en d'autres termes si vous avez une requête T-SQL que vous écrivez encore et encore, il vous suffit de créer une procédure stockée et de l'appeler). Les

procédures ressemblent à des constructions d'autres langages de programmation, car elles peuvent :

- Accepter des paramètres d'entrée et retourner plusieurs valeurs sous la forme de paramètres de sortie au programme appelant ;
- Contenir des instructions de programmation qui effectuent des opérations dans la base de données.

Les procédures stockées présentent les avantages suivants :

- Elles peuvent être facilement modifiées : Il est possible de modifier le code d'une procédure stockée sans avoir besoin de redémarrer ou de déployer l'application ;
- Elles garantissent une sécurité dans la mesure ou elles réduisent de potentielles menaces en éliminant l'accès direct aux objets et il est également possible de chiffrer des procédures stockées et de les déchiffrer grâce à des outils tiers (ApexSQL DECRYPT).
- Elles garantissent un temps de réponse plus rapide et une meilleure performance du système ;
- Elles évitent de réécrire plusieurs fois les mêmes instructions ;
- Il est possible de donner aux utilisateurs le droit d'exécuter une procédure stockée sans qu'ils aient le droit sur les objets qu'elle manipule.

a. Les types de procédures stockées

- **Définies par l'utilisateur :** une procédure définie par l'utilisateur peut être créée dans une base de données définie par celui-ci ou toute autre base de données système, à l'exception de la BD *resource*.
- Les procédures temporaires: elles sont une forme de procédure définie par l'utilisateur. Il en existe deux types: les procédures temporaires locales et les procédures temporaires globales, elles se différencient par leur nom, leur visibilité et leur disponibilité. Le premier caractère du nom des procédures temporaires locales est un # unique. Ces procédures sont visibles uniquement à la connexion actuelle de l'utilisateur et sont supprimées dès que la connexion est fermée.

Par contre, le nom des procédures temporaires globales commence par ##. Ces procédures sont visibles à tout utilisateur après avoir été créées st sont supprimées à la fin de la dernière session qui utilise la procédure.

Les procédures systèmes : elles sont incluses dans SQL Server et elles sont stockées physiquement dans la base de données *resource*, mais elles apparaissent logiquement dans le schéma sys de chaque base de données définie par le système et par l'utilisateur. La base de données msdb contient également les procédures stockées système dans le schéma dbo, utilisées pour planifier les alertes.

NB:

- Étant donné le fait que les procédures stockées systèmes commencent par le préfixe sp_ il est recommandé de ne pas utiliser ce préfixe quand vous nommez vos procédures ;
- Si un utilisateur crée une procédure stockée portant le même nom qu'une procédure stockée système, cette procédure ne s'exécutera jamais ;
- La programmation d'une procédure stockée diffère selon que cette procédure ne reçoit aucun paramètre, reçoit des paramètres en entrée, renvoie des paramètres de sortie ou retourne une valeur.

b. Créer une procédure stockée

Sous SSMS:

Pour créer une procédure dans l'explorateur d'objet :

- i. Connectez-vous à une instance du moteur de BD et développez là.
- ii. Développez Base de données, sélectionnez votre base de données ensuite cliquez sur Programmabilité.
- iii. Cliquez avec le bouton droit sur **Procédures stockées**, puis cliquez sur **Nouvelle procédure stockée**.
- iv. Dans le menu Requête, cliquez sur Spécifier les valeurs des paramètres du modèle.
- v. Dans la boite de dialogue, spécifiez les valeurs des paramètres du modèle. Entrez les valeurs suivantes :

Paramètres	Valeurs
Auteur	Votre nom
Date de création	Date du jour

Description	Données qui retournent ce que fait votre procédure
D 1	1
Procedure_name	Nom de la procédure stockée
@param1	@nomduparametre
@datatype_form_param1	Type du paramètre 1
Default_Value_form_param1	Valeur par défaut du paramètre 1
@param2	@nomduparametre2
Default_Value_form_param2	Valeur par défaut du parametre 2
@datatype_form_param2	Type du paramètre 2

- vi. Cliquez sur **OK**.
- vii. Dans l'éditeur de requête, écrivez votre requête;
- viii. Pour tester la syntaxe dans le menu **Requête**, cliquez sur **Analyser**. Si un message d'erreur est retourné, comparez les instructions avec les infos proposées et apportez les corrections nécessaires.
- ix. Pour exécuter la procédure dans le menu **Requête**, cliquez sur **Exécuter**.
- x. Pour exécuter la procédure, cliquez sur le bouton droit sur le nom de la procédure et sélectionnez **Exécuter la procédure stockée**. Et enfin entrez les paramètres correspondants.

XIV- Utilisation de T-SQL:

Syntaxe

CREATE PROCEDURE schema.nomprocedure

@param1 type,

@param2 type

AS

SET NOCOUNT ON; // empêche l'affichage du nombre de lignes affectées par l'instruction réduisant dc ainsi le temps de réponse sur le trafic

//instructions

Exemple : créez une procédure stockée qui prend en entrée le prénom d'un étudiant et affiche son nom

CREATE PROCEDURE gl.affichenom

@prenom nvarchar (20)

AS

SET NOCOUNT ON:

SELECT nom FROM DBO.ETUDIANT WHERE prenom = @prenom

GO

Devoir:

- 1. Procédure stockée qui affiche le nombre d'étudiants inscrits
- 2. Procédure stockée qui prend en paramètre le matricule d'un étudiant puis affiche sa classe et son niveau.

c. Exécution d'une procédure

Exec schema.nomdelaprocedure ; ou Exec schema.nomprocedure @param = " "

d. Déclaration d'une variable

Declare @nomvariable1 type, ou declare @nomvariable1 type, @nomvariable2 type;

e. Affectation d'une valeur à une variable

Select @nomvariable = valeur; ou Set @nomvariable = valeur;

f. Affichage d'informations

Print 'elementsaafficher'; print @nomvariable

g. Procédure stockée avec paramètre de sortie

Une Procédure stockée avec paramètre de sortie renverra une valeur après l'exécution.

Syntaxe:

CREATE PROCEDURE schema.nomprocedure

@param1 type,

@param2 type output

AS

SET NOCOUNT ON; // empêche l'affichage du nombre de lignes affectées par l'instruction réduisant dc ainsi le temps de réponse sur le trafic //instructions GOCependant, l'exécution d'une telle procédure est un peu différente, il faut d'abord déclarer une variable qui contiendra le paramètre de sortie renvoyé par la procédure. Exemple : créer une procédure stockée qui insere le matricule, le nom, l'idclass et la date de naissance puis retourne l'age de l'etudiant. CREATE PROCEDURE gl.insval_student @Smatricule varchar(50), @Snom varchar(50), @Sidclass int, @Sdatenaiss date, @Sage int output AS SET NOCOUNT ON; INSERT INTO gl.etudiant (matricule,nom,idclass,datenaiss) VALUES (@Smatricule,@Snom,@Sidclass,@Sdate); Set @Sage = select year(getdate()) - year(datenaiss) from gl.etudiant where matricule = @Smatricule; Exécution: Declare @Sage int

```
Exec gl.insval student @matricule = 'GL100', @nom = 'belinga', @idclass= 3, @datenaiss=
'1984-03-22', @Sage = @age output
Select @Sage
   h. Utilisation des structures alternatives
                ♣ If ...else
If condition
      Begin
            //instructions
      End
Else
      Begin
            //instructions
      End
Exemple: Procédure qui affiche un message d'erreur à l'utilisateur si le prenom renseigné ne se
trouve pas dans la base de données
CREATE PROCEDURE dbo. afficheprenomaveerreur
@prenom nvarchar (20)
AS
SET NOCOUNT ON;
DECLARE @prenom1 nvarchar (20);
SET @prenom1 = (SELECT prenom FROM dbo. Etudiants WHERE prenom=@prenom)
IF @prenom1 is null
      BEGIN
             PRINT 'l'étudiant n'existe pas'
      END
ELSE
      BEGIN
             SELECT nom, matricule FROM dbo. Etudiants WHERE prenom = @prenom1
      END
GO
```

Devoir:

On souhaite vérifier si l'étudiant Assoumou a une note en SQL SERVER si ce n'est pas le cas afficher : l'étudiant Assoumou n'a pas de note.

Case

Case

When condition 1 Then resultat 1

. . . .

When condition n Then resultat n

Else resultat n

End

Exemple : affiche la liste des étudiants (nom et prénoms) avec en plus une colonne genre qui affiche masculin, féminin ou autres en fonction de son sexe

CREATE PROCEDURE dbo. Affichegenre

AS

SET NOCOUNT ON;

SELECT nom, prenom, 'genre'=

CASE

WHEN sexe = 'M' THEN 'MASCULIN'

WHEN sexe = 'F' THEN 'FEMININ'

ELSE 'AUTRES'

END

FROM dbo.etudiant

GO

<u>Devoir</u> : affiche la liste des étudiants et qui indique pour chaque étudiant dans colonne type s'il est mineur ou majeur.

i. Utilisation des structures répétitives

WHILE condition

BEGIN

//instruction

END

END

Exemple : Tant que la moyenne d'âge des directeurs est inférieure à 30 ans, augmenter d'un an tous les directeurs qui ont un âge inferieur à la moyenne puis afficher à chaque fois la nouvelle liste des directeurs.

```
WHILE ((SELECT AVG(age) FROM dbo. directeur) < 30)

BEGIN

UPDATE dbo. directeur

SET age = age + 1

WHERE (age < (SELECT AVG(age) FROM dbo. directeur)

SELECT * FROM dbo. directeur

END
```

Tant que la moyenne d'âge des étudiants est inférieure à 30 ans, augmenter d'un an l'âge de tous les étudiants qui ont un âge inferieur à la moyenne puis afficher à chaque fois la nouvelle liste des étudiants.

```
Declare @moy int;

Select @moy = avg(year(getdate())- year(datenaiss)) from gl.Etudiant;

WHILE. @moy < 30

BEGIN

UPDATE gl.etudiant

SET datenaiss = dateAdd(year, -1, datenaiss)

WHERE year(getdate())- year(datenaiss) < @moy FROM gl.Etudiant;

SELECT * FROM gl.Etudiant;

Select @moy = avg(year(getdate())- year(datenaiss)) from gl.Etudiant;
```

Exemple 2: jusqu'à ce que la plus grande note en oracle soit 18, ajoutez deux points aux étudiants en cette matière, puis affichez à chaque fois la nouvelle liste des étudiants avec leur note en oracle.

La gestion des transactions

Une transaction est un bloc d'instructions DML exécutés et qui laisse la base de données dans un état cohérent. Si une seule instruction dans le bloc n'est pas cohérente ou si pour une raison quelconque l'une des instructions n'a pas été exécutées, tout le bloc est annulé. (prendre le cas du guichet automatique, demandez les requêtes qui interviennent)

- Propriétés d'une transaction : Toutes les transactions ont la propriété ACID
- A. Atomicité. Soit toutes les modifications sont effectuées, soit aucune ne l'est.
- C : Cohérence : Lorsqu'une transaction a débuté, celle-ci se déroule correctement jusqu'à la fin, dans le cas contraire le SGBD retourne la bd dans l'état ou elle était avant les modifications conformément aux règles, contraintes etc.
- I : isolement : Les modifications effectuées par des transactions concurrentes doivent être isolées transaction par transaction. (Cette fonctionnalité est assurée par les verrous).
- D : Durabilité : Dès qu'une transaction est terminée, ses effets sur le système sont permanents.

On distingue les principales transactions suivantes :

- Les transactions en mode validation automatique : chaque instruction individuelle est une transaction, celle-ci est validée ou annulée en fonction de son résultat ;
- Les transactions explicites: Chaque transaction est explicitement lancée par l'instruction BEGIN TRANSACTION et explicitement terminée par une instruction COMMIT ou ROLLBACK;
- les transactions implicites: Une nouvelle transaction est implicitement lancée lorsque la transaction précédente est terminée mais celle-ci est explicitement terminée par une instruction COMMIT ou ROLLBACK.

NB:

- Pour démarrer une transaction on utilise l'instruction BEGIN TRAN : indique le point de départ d'une transaction explicite

- Pour valider la transaction, et rendre les traitements qui lui sont associés effectifs, on utilise l'instruction COMMIT TRAN : elle a pour effet de valider définitivement les modifications apportées à la base de données
- Pour interrompre une transaction en cours qui n'a pas encore été validée, on utilise l'instruction ROLLBACK TRAN : restaure une transaction au début de cette transaction, aucune modification apportée à la transaction n'est validée dans la base de données
- SAVE TRANSACTION : diviser une transaction en plusieurs unités afin que l'utilisateur ait une chance de restaurer une transaction à partir d'un point de sauvegarde. Le point de sauvegarde insère une marque spéciale dans une transaction qui nous permet d'annuler toutes les modifications exécutées après le point de sauvegarde. Il est également utilisé pour annuler une partie spécifique de la transaction plutôt que la totalité de la transaction

N.B:

- Une fois, le commit effectué, nous ne pouvons pas Rollback les données
- Les transactions ne concernent que des requêtes de type LMD
 Exemples :

```
BEGIN TRAN exemple1
INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567',"BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02',1)
UPDATE GL.ETUDIANT
SET nom='NGUIAMBA',
   prenom= 'ANDRE'
WHERE matricule ='GL.Y4567'
DELETE FROM GL.ETUDAINT WHERE matricule ='GL.Y4567'
COMMIT TRAN exemple1
BEGIN TRAN exemple2
INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567', "BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02', 1)
INSERT INTO GL.ETUDIANT VALUES ('GL.Y4568',"AMOUGOU', 'FREDDDY', 'BARRIERE','1987-12-02',1)
UPDATE GL.ETUDIANT
SET nom='NGUIAMBA',
   prenom= 'ANDRE'
WHERE matricule ='GL.Y4567'
COMMIT TRAN exemple2
```

BEGIN TRANSACTION exemple 3

INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567', "BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02', 1) INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567', "BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02', 1)

IF(@@ERROR > 0)
BEGIN

Rollback Transaction exemple3

END

ELSE

BEGIN

Commit Transaction exemple3

END

BEGIN TRAN

INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567', "BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02', 1)

SELECT * FROM GL.ETUDIANT

ROLLBACK TRAN

SELECT * FROM GL.ETUDIANT

BEGIN TRANSACTION Exemple5

INSERT INTO GL.ETUDIANT VALUES ('GL.Y4567', 'BELINGA', 'ESTELLE', 'BARRIERE', '1987-12-02', 1)

SAVE TRANSACTION InsertStudent

DELETE FROM GL.ETUDAINT WHERE matricule ='GL.Y4567'

SELECT * FROM GL.Etudiant

ROLLBACK TRANSACTION InsertStudent

SELECT * FROM GL.Etudiant

j. La gestion des messages d'erreurs

Lors de chaque erreur rencontrée, le moteur associe à celle-ci un message. Si l'erreur rencontrée lors de l'exécution n'existe pas dans la table Sys. messages de la base de données Master elle obtient le code 50000.

Le moteur ajoute aussi une notion de sévérité (degré de gravité d'une erreur) qui est notée par une valeur :

Sévérité	Description
0-9	Les messages d'informations signalent des erreurs
	qui ne sont pas très graves
10	Les messages d'informations signalent des erreurs
	peu graves, le moteur de base de données convertit
	une gravité 10 en gravité 0

11-16	Indique des erreurs pouvant être corrigées par
	l'utilisateur
11	Indique que l'objet ou l'entité donnée n'existe pas
12	Utilisée pour des requêtes qui n'utilisent pas de
	verrous
13	Indique des erreurs de blocage de transaction
14	Indique des erreurs liées à la sécurité, telles qu'une
	autorisation refusée
15	Indique des erreurs de syntaxe
16	Indique des erreurs générales
17-19	Indique des erreurs logicielles qui ne peuvent pas
	être corrigées par l'utilisateur
17	Indique que l'instruction a provoqué une saturation
	des ressources de SQL server
18	Indique un problème dans le logiciel du moteur de
	base de données mais l'exécution de l'instruction
	est terminée et la connexion à l'instance du moteur
	de base de données est conservée. L'administrateur
	système doit être informé chaque fois qu'un
	message de gravité 18 est émis
19	Indique qu'une limite non configurable du moteur
	de base de données a été dépasée et que le
	processus de traitement par lots en cours a été
	arreté
20-24	Indique les problemes système et les erreurs fatales
	qui peuvent endommagées la base de données
20	Indique qu'une instruction a rencontré un problème
21	Indique qu'un problème a été rencontré et affecte
	toutes les taches de la base de données
22	Indique que la table ou l'index spécifié dans le
	message a été endommagé par un problème
	matériel ou logiciel

23	Indique que l'intégrité de la bd est en cause à cause
	d'un problème matériel ou logiciel
24	Indique une défaillance du système.
	L'administrateur système devra alors restaurer la
	base de données

L'instruction Raiseerror affiche un message d'erreur système

Syntaxe

Raiseerror(nummessage|message,gravité,état,parametres)

Nummessage : indique le numéro du message pour faire appel à un message déjà disponible dans la table sys.Messages

Message: votre message à afficher

Gravité :représente le niveau de gravité.Seul l'administrateur système peut ajouter des messages avec un niveau de gravité compris entre 19 et 25

Etat : valeur comprise entre 0 et 255 qui identifie la source à partir de laquelle l'erreur a été émise Parametres :parametres servant à la substitution des variables définies dans le message, ces paramètres ne peuvent etre que de type int, varchar, binary ou varbinary

18.Les Vues

Une vue est une table virtuelle dont le contenu est le résultat d'une requête. Les lignes et les colonnes d'une vue proviennent des tables référencées dans la requête. Les vues sont très utiles dans la mesure où elles permettent de simplifier et personnaliser la perception de la base de données et peuvent être utilisées comme mécanisme de sécurité en permettant aux utilisateurs d'accéder aux données par le biais des vues sans leur accorder d'autorisations qui leur permettraient d'accéder directement aux tables.

N.B:

- La suppression d'une vue n'entraine pas la suppression des données
- Une vue ne peut faire référence qu'à un maximum de 1024 colonnes

- Les clauses SELECT faisant partie d'une définition d'une vue ne peuvent inclure la clause ORDER BY sauf si une clause TOP figure également dans la liste de sélection de l'instruction SELECT
- Si une vue dépend d'une table ou d'une vue qui a été supprimée, le système renverra un message d'erreur.

a. Créer une Vue

Créer une vue qui affiche les étudiants (noms, prenoms, date de naissance) ainsi que leur salle de classe

Use Gl; Go

Create View gl. Vue Etudiant

As

Select e.Nom, e.prenom, e.datenaiss, s.nomsalle

From gl.etudiant e inner join gl.salle s

On e.idsalle = s.idsalle;

go

---- affichage : select * from gl.VueEtudiant ;

b. Modifier une vue

Modifier la vue VueEtudiant, affichant les informations des étudiants nés avant 2001 Alter View gl.VueEtudiant

As

Select e.Nom, e.prenom, e.datenaiss, s.nomsalle

From gl.etudiant e inner join gl.salle s

On e.idsalle = s.idsalle Where year(datenaiss) < 2001;

c. Supprimer une vue

Drop View gl. Vue Etudiant;

d. Créer une vue partitionnée avec UNION ALL

Une vue partitionnée joint les données partitionnées de manière horizontale d'un ensemble de tables membres sur un ou plusieurs serveurs ; les données sont ainsi affichées comme si elles provenaient d'une seule table. Une vue qui joint des tables sur la même instance de SQL SERVER est appelée vue partitionnée locale.

Syntaxe : (on crée une vue des étudiants de gl qui joint des données des tables des étudiants gl de yde, dla, bafoussam, ombessa)

CREATE VIEW gl. VueEtudiant

AS

SELECT matricule,nom,prenom,sexe,quartier,datenaiss,idclasse FROM gl.Etudiant yde

UNION ALL

SELECT matricule,nom,prenom,sexe,quartier,datenaiss,idclasse FROM gl.Etudiant dla

UNION ALL

SELECT matricule,nom,prenom,sexe,quartier,datenaiss,idclasse FROM gl.Etudiant_baf UNION ALL

SELECT matricule,nom,prenom,sexe,quartier,datenaiss,idclasse FROM gl.Etudiant ombess

N.B : Les tables membres doivent obligatoirement avoir la même structure.

IIIe Partie: Exposés

<u>Thème 1</u>: Gestion des authentifications et autorisations

- Principe
- Présenter les différents rôles systèmes
- Création des utilisateurs
- Création des comptes de connexion
- Autorisation des comptes de connexion pour accéder à une base de données
- Autorisation entre serveurs
- Autorisation des utilisateurs à accéder aux objets et aux objets d'un schéma
- Autorisation des utilisateurs à exécuter du code

<u>Thème 2</u>: Sauvegarde des bases de données sur SQL server

- Type de sauvegardes
- Stratégies de sauvegarde
- Présenter le fonctionnement du journal de transaction
- Planification d'une stratégie de sauvegarde
- Processus de sauvegarde de bases de données et des journaux de transaction
- Comment se fait la gestion des sauvegardes

<u>Thème 3</u>: Restauration des bases de données sur SQL SERVER

- Type de restauration
- Présenter le processus de restauration
- Comment restaurer les bases de données systèmes et de fichiers individuels

<u>Thème 4</u>: Surveillance de SQL server à l'aide d'alertes et de notifications

- Configurer la messagerie de base de données
- Surveillance des erreurs SQL server
- Configuration des opérateurs, alertes et notifications

- Surveillance de l'activité
- Capturer et gérer les données de performance
- Analyser les données de performance collectées

Thème 5 : Import et export des données

- Comment se fait le transfert de données depuis SQL server
- Comment se fait l'import et l'export des données des tables
- Utilisation BULK INSERT, Openrowset, bcp , select into
- Utilisation de ApexSQL SCRIPT

<u>Thème 6</u>: Traçage de l'accès

- Principe
- Capture de l'activité à l'aide de SQL server profiler
- Comment améliorer les performances avec l'assistant de paramétrage de base de données
- Comment travailler avec les options de traçage
- Surveillance des verrous

<u>Thème 7</u>: Utilisation de SQL SERVER reporting service

- Configuration
- Fonctionnement
- Architecture
- Outils nécessaires pour l'utilisation de ce composant
- Présenter le processus de reporting
- Présenter différents rapports (visualisation)

Thème 8 : Analyse des données avec SQL server Analysis service

- Principe
- Configuration
- architecture
- Cubes olap, dimensions, des tables de fait
- Création d'un cube olap
- Comment explorer les données du cube

Thème 9 : Les déclencheurs

Critères d'évaluation:

- Rapport physique /5
- Présentation orale /5
- Réponse aux questions et compréhension du thème /5
- Démo /5

WEBOGRAPHIE

https://docs.microsoft.com

https://webdevdesigner.com/q/what-is-the-difference-between-sql-pl-sql-and-t-sql-3130/

https://www.oracle.com

https://www.lebigdata.fr/olap-online-analytical-processing

https://docs.microsoft.com/fr-fr/sql/sql-server/install/sql-server-components

https://www.supinfo.com/articles/single/6498-sql-server-architectures-structure-une-base-

donnees

https://www.supinfo.com/articles/single/3522-cube-olap