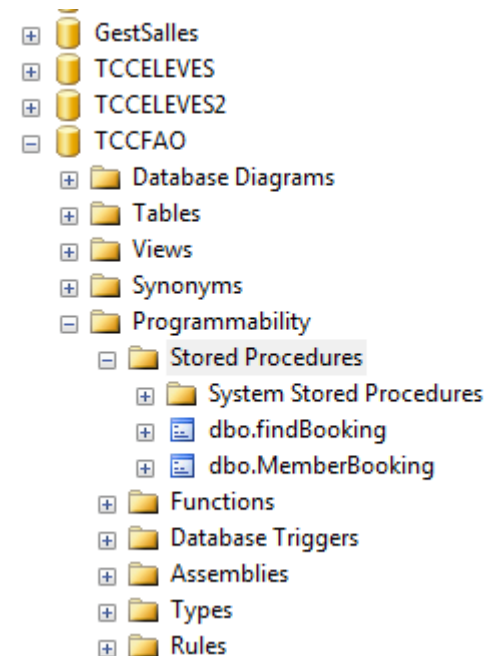


PROCÉDURES STOCKÉES ET FONCTIONS

Procédures stockées

- Série d'instructions
- Stockées dans la BD
- Font partie intégrante de la BD



Syntaxe

```
CREATE PROCEDURE <nom proc> [param_name  
param_type] [ ,...n ] AS  
BEGIN  
    declare @<var_name> <type>  
    corps de la procédure  
END
```

Plus de paramètres sur <https://msdn.microsoft.com/fr-ch/library/ms187926.aspx>

Pour supprimer la procédure :

```
DROP PROCEDURE <nom proc>
```

Exemples

```
CREATE PROCEDURE findAllBooking  
AS  
BEGIN  
SELECT *  
FROM booking  
END
```

```
CREATE PROCEDURE findBooking @idUser int, @idPartner  
int  
AS  
BEGIN  
SELECT *  
FROM booking  
WHERE fkMadeBy = @idUser and fkPartner = @idPartner  
END
```

Comment exécuter ces procédures?

Exec <procedure name>

ou

Execute <procedure name>

Exemples :

exec findAllBooking

exec findBooking 59, 138

Exemple de procédures stockées

Procédures stockées fournies dans SQL Server :

Présentes dans la base « master »

Commencent par sp_

Exemple :

sp_help booking

sp_helptrigger booking

Les 3 types de fonctions

Fonction qui retourne une valeur scalaire

```
CREATE FUNCTION function_name ( [ @param_name  
param_type [ = default ] [ ,...n ] ] ) RETURNS return_type  
AS  
BEGIN  
    function_body  
    RETURN valeur  
END
```

Exemple

```
CREATE FUNCTION getNbBookings() RETURNS int
AS
BEGIN
    declare @nb int;
    select @nb = count(*) from booking;
    return @nb;
END
```


Les 3 types de fonctions

Fonction qui retourne une table en ligne

```
CREATE FUNCTION function_name ( [ @param_name  
param_type [ = default ] [ ,...n ] ] ) RETURNS TABLE AS  
RETURN  
    SELECT_STATEMENT
```

Example

```
CREATE FUNCTION getAllBookings() RETURNS table
AS
return (
    select * from booking
)
```

Les 3 types de fonctions

Fonction qui retourne une table multi-instructions

```
CREATE FUNCTION function_name ( [ @param_name  
param_type [ = default ] [ ,...n ] ] )  
RETURNS @variable_retour TABLE (nom_colonne type,...)  
AS  
BEGIN  
    function_body  
    RETURN;  
END
```

Exemple

```
CREATE FUNCTION AnalyseClient (@nocli int)
RETURNS @FicheClient TABLE(libelle varchar(30), valeur int)
AS
BEGIN
    INSERT INTO @FicheClient
    VALUES ('Nombre de commandes',dbo.nbre_cde(@nocli));
    INSERT INTO @FicheClient
        SELECT 'Montant total',
            CONVERT(int,SUM(qte_cde*prixht_art))
        FROM COMMANDES cde INNER JOIN LIGNES_CDE lig
        ON cde.numero_cde=lig.numero_cde
        INNER JOIN ARTICLES art
        ON lig.reference_art=art.reference_art;
    RETURN;
END;
```

Comment exécuter ces fonctions?

Exemples :

```
select * from getAllBookings()  
select dbo.getNbBookings()  
select * from AnalyseClient(1)
```

Pourquoi des procédures / fonctions?

- Gestion des permissions plus fine => permission sur la procédure mais pas sur les tables sous-jacentes
- Centralisation du code dans la base de données si utilisée par plusieurs applications