

EXPOSE DE PROJET BD

# LES PROCEDURES STOCKEES

Par:

BAYOI Michel  
MBALLA Albert  
ASSIENE James  
JIOKENG Kevin

Sous la supervision de: Dr NANA MBINKEU

# PLAN DE L'EXPOSE

## INTRODUCTION

### I. DÉFINITION ET AVANTAGES DE PROCÉDURES STOCKÉES

### II. CODIFICATION DES PROCÉDURES STOCKÉES

1. Création des procédures stockées
2. Modification des procédures stockées
3. Suppression des procédures stockées
4. Appel d'une procédure stockées
5. Définition des paramètres

### III. EXEMPLES

## CONCLUSION



# INTRODUCTION

L'importance des bases de données n'est plus à démontrer. Mais bien souvent on doit répéter les mêmes instructions avec quelques fois des paramètres différents. Les procédures stockées viennent alors simplifier la tâche au programmeur.



# I. Définition et avantages de procédures stockées

# Définition de procédures stockées

- o Une procédure stockée est une collection précompilée d'instructions Transact-SQL stockée sous un nom et traitée comme une unité.
- o Elles renvoient les données de quatre façons différentes. Mais nous nous intéressons ici à deux façons:
  - des paramètres de sortie, qui renvoient soit des données (entiers ou caractères) ou une variable de curseur, les curseurs étant des ensembles de résultats pouvant être extraits ligne par ligne ;
  - un ensemble de résultats pour chaque instruction SELECT contenue dans la procédure stockée ou toute autre procédure stockée appelée par cette dernière ;



# Avantages des procédures stockées

- o Les procédures stockées contribuent à mettre en œuvre une logique cohérente dans les applications. Les instructions SQL et la logique nécessaire à l'exécution d'une tâche fréquente peuvent être créées, codées et testées une seule fois dans une procédure stockée.
- o Les procédures stockées peuvent également améliorer les performances.
- o Les procédures stockées évitent aussi aux utilisateurs d'avoir à connaître les détails des tables de la base de données



## II. CODIFICATION DES PROCEDURES STOCKEES

# 1. Création des procédures stockées

- o La syntaxe pour créer une procédure stockée est la suivante:

```
CREATE PROC [EDURE] procedure_name [ ; number  
]  
    { @parameter data_type }  
AS sql_statement [ ...n ]
```



## 2. Modification des procédures stockées

- Pour modifier une procédure stockée on utilise la syntaxe suivante:

```
ALTER PROC [EDURE] procedure_name [ ; number ]  
    { @parameter data_type }  
    AS sql_statement [ ...n ]
```

### 3. Suppression de procédures stockées

- o La suppression de procédures stockées se fait de comme suit:

**DROP PROC [EDURE] procedure\_name**

## 4. Appel d'une procédure stockée

- On utilise l'instruction CALL ou EXECUTE.  
(EXEC[UTE] | CALL ) procedure\_name [@parameter 1 = value1[, @parameter2 = value2[, @parameter3 = value3...]]]
- Des valeurs de paramètres peuvent être fournies si une procédure stockée a été écrite pour les accepter.



## 4. Appel d'une procédure stockée (2)

- o Si vous entrez des paramètres sous la forme *@Parameter = value*, leur ordre n'a pas d'importance. Vous pouvez aussi omettre les paramètres pour lesquels une valeur par défaut a été définie
- o Si vous spécifiez un paramètre donné sous la forme *@Parameter = value*, vous devez tous les spécifier de cette façon. Sinon, ils doivent apparaître dans l'ordre indiqué par l'instruction CREATE PROCEDURE.
- o Pour faire appel à une procédure stockée on utilise le mot-clé **CALL**

## 5. Définition des paramètres

Pour déclarer un paramètre il suffit de spécifier dans l'entête de la procédure:

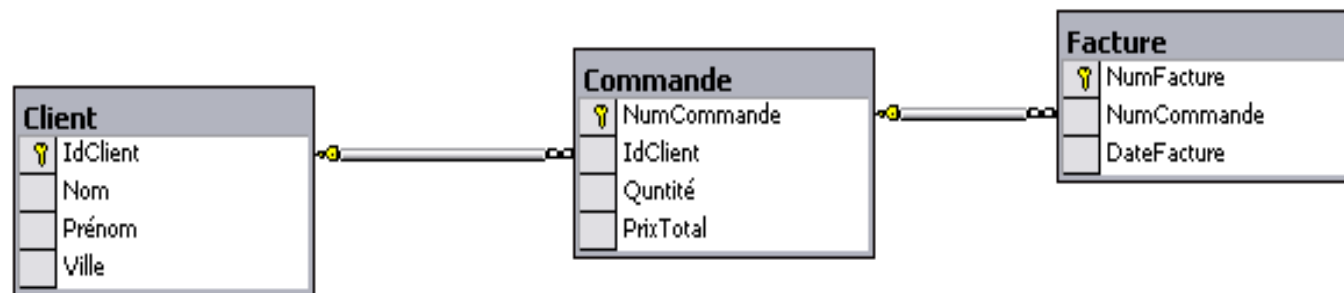
- Son sens : entrant, sortant, ou les deux.
  - **IN** pour entrant
  - **OUT** pour sortant
  - **INOUT** pour les deux
- Si aucun sens n'est donné, il s'agira d'un paramètre IN par défaut.
- Son nom : indispensable pour le désigner à l'intérieur de la procédure.
- Son type : INT, VARCHAR(10)

### III. EXAMPLES



# Exemples

Dans les exemples qui suivent, nous travaillons avec la base de données « GestionCommande »:



# Exemples (2)

----Création d'une procédure stockée simple

```
Create procedure PS1
```

```
as
```

```
select * from Client
```

```
where IdClient>1650
```

----Exécution

```
CALL PS1
```

----Création d'une procédure avec un seul paramètre

```
Create proc myprocedure
```

```
@d datetime
```

```
as
```

```
select * from Facture where DateFacture=@d
```

----Exécution

```
CALL myprocedure '08/09/2002'
```

# Exemples (3)

----Procédure qui a un paramètre optionnel

```
Create proc PS;4
```

```
(@Id int =null)
```

```
As
```

```
If @Id is null
```

```
Begin
```

```
Select * from Client
```

```
End
```

```
Else Begin
```

```
Select * from Client where IdClient=@Id
```

```
End
```

```
Go
```

----Exécution

```
CALL PS;4 @Id=1925
```

```
CALL PS;4 1668
```

```
CALL PS;4
```

```
CALL PS;4 @Id=default
```



# CONCLUSION

Les procédures stockées permettent de regrouper certaines opérations qui pourront être exécutées plusieurs fois. Comme les fonctions dans les langages de programmation procédurale, elles sont indubitablement d'une utilité essentielle.