

# Stored Procedures

A stored procedure is:

- A set of precompiled SQL statements
- Stored in the database, on the server
- Executable on request, at any time

# Stored Procedures

## Interests

- Increased execution speed:
  - Server storage → less client-server exchanges
  - Precompiled
- Security: Interfacing between the user and the tables
- Readability: Using a programming language

## Syntaxe

```
CREATE PROC <name_proc> [ ( <list_params> ) ] AS  
BEGIN
```

```
    <instructions>
```

```
END
```

<liste\_params> : Comma separated list of parameters

Parameter: @<param> <type> [= <valeur\_défaut>]

# Simple example

Declaration and definition of the stored procedure:

```
IF EXISTS (SELECT * FROM sys.procedures WHERE
                                                NAME='mySimpleProc')
    DROP PROC mySimpleProc
GO

CREATE PROCEDURE mySimpleProc (@param int) AS
BEGIN
    PRINT 'Parameter: ' + CONVERT(VARCHAR, @param)
    SET @param = @param + 1
    PRINT 'I return ' + CONVERT(VARCHAR, @param)
    RETURN @param
END
GO
```

# Simple example

Simple call :

```
EXEC mySimpleProc 54
```

Result:

```
'Parameter: 54  
I return 55
```

Call with print:

```
DECLARE @resultat INT  
EXEC @resultat=mySimpleProc 54  
PRINT @resultat
```

Résultat :

```
Parameter: 54  
I return 55  
55
```

## Advanced example

Declaration and definition of the stored procedure:

```
IF EXISTS (SELECT * FROM sys.procedures WHERE
                                                NAME='anotherProc')
    DROP PROC anotherProc
GO
```

```
CREATE PROCEDURE anotherProc (@auteur VARCHAR(20)) AS
BEGIN
    SELECT au_fname + ' ' + au_lname AS 'Nom',
           title AS 'Titre'
    FROM titles T JOIN titleauthor TA ON T.title_id=TA.title_id
                JOIN authors A ON TA.au_id=A.au_id
    WHERE A.au_lname LIKE @auteur
END
GO
```

# Advanced example

Uses:

EXEC **anotherProc** 'Ringer'

Results:

	Nom	Titre
1	Albert Ringer	Is Anger the Enemy?
2	Albert Ringer	Life Without Fear
3	Anne Ringer	The Gourmet Microwave
4	Anne Ringer	Is Anger the Enemy?

EXEC **anotherProc** '[BD]%'

	Nom	Titre
1	Abraham Bennet	The Busy Executive's Database Guide
2	Reginald Blotchet-Halls	Fifty Years in Buckingham Palace Kitchens
3	Michel DeFrance	The Gourmet Microwave
4	Innes del Castillo	Silicon Valley Gastronomic Treats
5	Ann Dull	Secrets of Silicon Valley

# More advanced example

-- drop table if exists

```
IF OBJECT_ID('whileTab') IS NOT NULL DROP TABLE whileTab
GO
```

-- create test table whileTab '

```
CREATE TABLE whileTab (
    id      INT IDENTITY PRIMARY KEY,
    nom     VARCHAR(20),
    date    DATETIME
)
GO
```

-- drop the stored procedure if exists

```
IF OBJECT_ID('addLines') IS NOT NULL DROP PROC addLines
GO
```

-- define the stored procedure

```
CREATE PROC addLines(@nbLignes int, @nomBase VARCHAR(20) = 'default') AS
    WHILE (@nbLignes > 0)
    BEGIN
        INSERT INTO whileTab(nom, date)
            VALUES (@nomBase+CONVERT(VARCHAR(20), @nbLignes), GETDATE())
        SET @nbLignes -= 1
    END
GO
```



# More advanced example

Usage:

```
EXEC addLines 5
```

```
EXEC addLines 3, 'toto'
```

```
GO
```

```
SELECT * FROM whileTab
```

Result:

	id	nom	date
1	1	default5	2011-01-11 08:55:59.673
2	2	default4	2011-01-11 08:55:59.673
3	3	default3	2011-01-11 08:55:59.673
4	4	default2	2011-01-11 08:55:59.673
5	5	default1	2011-01-11 08:55:59.673
6	6	toto3	2011-01-11 08:55:59.673
7	7	toto2	2011-01-11 08:55:59.673
8	8	toto1	2011-01-11 08:55:59.673