# Les procédures stockées

# Définition

- Une procédure stockée (Stored Procedure) est une suite d'instructions SQL stockées au niveau de la base de données.
- Une procédure stockée va apparaitre comme un d'objet de la base de données (au même titre qu'une table)
- Une fois la procédure crée on pourra l'appeler par son nom et donc les instructions de la procédures seront exécutées.

# Création d'une procédure stockée

- Syntaxe:

```
CREATE   PROCEDURE   nom_procedure (par1, par2,…)

Corps de la procedure;
```

- Le nom de la procédure peut être suivi de paramètre ou pas.
- Les parenthèses sont obligatoires
- Le corps de la procédures sont les instructions qui vont être exécutées par la procédure.

# Exemple de procédure

```
CREATE   PROCEDURE  ListeClient( )
Select * from clients;
```

- Si la procédure contient plusieurs instructions, celles-ci doivent être délimitées par **BEGIN** et **END.**

# Le délimiteur de la procédure

- L'exemple précédent on pourrait l'écrire comme suit:

```
CREATE   PROCEDURE  ListeClient()
BEGIN
Select * from clients ;
END ;
```

# Le délimiteur de la procédure

- Si on exécute la procédure précédente, une erreur va être affichée à cause du **; car le ;** c'est la fin des instructions, c'est comme si la commande CREATE  PROCEDURE a terminée.

- ce qui fait **begin** reste ouverte et déclenche une erreur.

- La solution est d'utiliser un délimiteur autre que ; comme | ou $...

# Le délimiteur de la procédure

- La procédure précédente va s'écrire donc comme suit:

```
DELIMITER |
CREATE PROCEDURE ListeClienst()
BEGIN
Select * from clients ;
END |
DELIMITER ;
```

# Appel d'une procédure stockée

- L'appel d'une procédure, pour l'exécuter, se fait par son nom en utilisant la commande **CALL:**

```
CALL   ListeCLients();
```

- Dans cet exemple le résultat de la procédure c'est le résultat de la requête select * from clients

# Les paramètres d'une procédure stockée

- Lorsque on passe un paramètre à une procédure, il faut définir son **orientation(direction)** et son type, comme suit:

```
Orientation   nomParamètre  Type
```

- Le type du paramètre c'est le type de mysql que vous connaissez (int, varchar….), l'orientation à voir par la suite.

# L'orientation d'un paramètre

- Un paramètre peut être de trois sens différents : entrant (**IN**), sortant (**OUT**), ou les deux (**INOUT**).
- **IN** : il s'agit d'un paramètre dont la valeur est fournie à la procédure stockée. Cette valeur sera utilisée durant l'exécution de la procédure.
- **OUT** : il s'agit d'un paramètre "sortant", dont la valeur sera établie au cours de la procédure.
- **INOUT** : un tel paramètre pourra être utilisé au même temps comme **IN** et **OUT**

# Procédure avec un paramètre entrant

- Exemple :

```
DELIMITER |
CREATE   PROCEDURE  ListeClient(IN code varchar(5))
BEGIN
Select * from clients  where `code client`=code;
END |
```

- Appel de la procédure:

```
CALL listeCLient('ALFKI');  ou bien

Set @x:='ALFKI';
CALL listeCLient(@x);
```

# Procédure avec un paramètre entrant et un sortant

- Exemple:

```
DELIMITER |
CREATE   PROCEDURE  ListeClient(IN code varchar(5),
OUT  soc varchar(20))
BEGIN
Select  societe INTO soc   from clients  where `code
client`=code;
END |
DELIMITER ;
```

# L'appel de la procédure avec paramètre de sortie

```
Set @x:='ALFKI';
CALL ListeClient(@x,@var);
SELECT @var;
```

- SELECT @var: c'est pour afficher la valeur du paramètre de sortie.

# Suppression d'une procédure

- La suppression d'une procédure stockée est simple:

```
DROP  PROCEDURE  NomDeProcedure;
```