

Bases de données
PL-SQL
MySQL – Procédures et fonctions stockées
Bertrand LIAUDET

SOMMAIRE

SOMMAIRE	1
PL-SQL - PROCEDURES ET FONCTIONS STOCKEES	3
1. PL-SQL : les procédures stockées	3
Présentation	3
Script d'exemple	4
Usage des procédures stockées : CALL	5
Gestion des procédures stockées	5
Gestion des erreurs	5
2. PL-SQL : Eléments de programmation	6
Afficher du texte	6
Afficher une table	6
Commentaires	6
Variables locales : déclaration, type et affectation	7
Variables globales : @	8
Paramètres en entrée : IN	9
Paramètres en sortie : OUT et INOUT	10
Utilisation de commandes du DDL, DML et DCL	11
Tous les types	12
Opérateurs et fonctions accessibles	14
Tests – IF – ELSE - ELSEIF	15
CASE WHEN	15
Boucles	16
Blocs imbriqués	18
Sortie de bloc : leave	18
Les curseurs	19
3. PL-SQL : Les fonctions stockées	21
Présentation	21
Usage des fonctions stockées	21
Gestion des fonctions stockées	21
Synthèse et méthode	22
Synthèse	22
Méthode de codage	22

TP PROCEDURES ET FONCTIONS STOCKEES	23
0. MySQL Workbench	23
Présentation	23
Téléchargement	23
Installation	23
Résultats de l'installation	23
Principes du fonctionnement	23
1. La bibliothèque – procédures et fonctions stockées	24
2. Les chantiers – procédure stockée	26
3. BD Ecoling - fonction stockée	26
4. Programmation classique – procédures stockées	27

Edition juillet 2015

PL-SQL - PROCEDURES ET FONCTIONS STOCKEES

1. PL-SQL : les procédures stockées

<http://dev.mysql.com/doc/refman/5.6/en/stored-procedures.html>

Présentation

Usages

Les SGBD-R en général, et MySQL en particulier, permettent d'écrire des procédures de programmation impérative classique (type Pascal, C, VB, PHP, etc.) qui seront enregistrées dans la base de données.

Ces procédures ont plusieurs usages :

➤ *Approche objet*

Ces procédures (et fonctions) permettent de développer des méthodes d'utilisation de la BD, méthode qui cachent un accès direct aux données. Elles jouent alors le rôle des méthodes dans les classes de la programmation objet.

Les procédures vont encapsuler les attributs et permettre certains usages particuliers.

➤ *Synthèse – approche décisionnel*

Les procédures stockées permettent de faire des calculs de synthèse sur les données de la BD. Elles permettent aussi de créer de nouvelles tables pour recevoir les résultats, si nécessaire.

➤ *Administration et archivage*

Les procédures stockées peuvent aussi servir pour gérer des tâches d'archivage récurrentes et pour des tâches d'administration des utilisateurs et/ou des données.

Principes généraux de codage

Dans une procédure PL-SQL, on peut utiliser les techniques de programmation procédurale classique : variable, test, boucle et fonction.

On peut aussi intégrer les ordres SQL.

Principes généraux d'optimisation

Il faut toujours, autant que possible, régler les problèmes avec des commandes SQL et éviter la programmation impérative classique : le résultat sera toujours plus optimisé ou plus optimisable.

Dans le même esprit, il faut toujours éviter d'utiliser des curseurs et utiliser les possibilités du SQL.

Script d'exemple

```
-- script de définition d'une procédure
-- procédure « insertemp » : permet d'insérer un employé avec :
-- son numéro, son nom et son numéro de département

use empdept;
drop procedure if exists insertemp;
delimiter //

create procedure insertemp (v_ne integer, v_nom varchar(14),
v_nd integer)
comment 'permet d'insérer un employé avec ses numéro, nom et n°
de dept'
begin
    insert into emp(ne, nom, nd) values (v_ne, v_nom, v_nd);
end ;
//
delimiter ;
```

Explications

- La procédure est créée avec l'instruction : « create procedure »
- Une liste de variable est passée en paramètre.
- Des commentaires sont ajoutés avec le « comment »
- Le corps de la procédure commence par « begin » et finit par « end ; »
- Dans le corps de la procédure on peut mettre des requêtes SQL et utiliser les paramètres de la procédure.
- Avant la création, il faut changer de délimiteur : delimiter //. Ceci vient du fait que la procédure utilise des « ; » comme délimiteur, et que le « ; » est le délimiteur standard du SQL.
- On termine la procédure par un délimiteur : //
- Après le //, il faut revenir au délimiteur standard : delimiter ;

Usage des procédures stockées : CALL

Le script précédent permet de créer la procédure « insertemp ».

Cette procédure s'utilise ainsi :

```
CALL insertemp (9500, « Durand », 10) ;
```

Le « call » permet d'appeler la procédure. On passe les paramètres à la procédure. L'instruction permet de créer un nouvel employé : Durant, n°9500 dans le département 10.

Gestion des procédures stockées

Afficher les procédures existantes :

```
Show procedure status ;
```

Afficher le code d'une procédure :

```
Show create procedure insertemp ;
```

Supprimer une procédure :

```
Drop procedure insertemp ;
```

Créer une procédure :

```
Create procedure insertemp...  
;
```

<http://dev.mysql.com/doc/refman/5.6/en/create-procedure.html>

Gestion des erreurs

Afficher les erreurs-warnings :

```
Show warnings ;
```

2. PL-SQL : Éléments de programmation

Afficher du texte

```
/* script de définition d'une procédure
   procédure « bonjour » : affiche bonjour,
   usage des commentaires en style C
*/

drop procedure if exists bonjour;
delimiter //
create procedure bonjour ( )
begin
    select 'bonjour';
    select 'tout le monde';
end ;
//
delimiter ;
```

```
mysql> call bonjour();
+-----+
| bonjour |
+-----+
| bonjour |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

Afficher une table

Principe

```
...
begin
    select * from emp;
end ;
//
delimiter ;
```

Commentaires

```
/* script de définition d'une procédure
   procédure « bonjour » : affiche bonjour,
   usage des commentaires en style C
*/

-- commentaires derrière deux tirets et un espace
```

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.

