



# PHP

## PDO PROCEDURES STOCKEES

# PRINCIPE

- **Bout de code SQL enregistré directement dans la base**
- **Sur-ensemble de SQL (*variables, sélections (if) boucle, curseur*)**  
*PL SQL (Oracle)*  
*T-SQL (MS SQL Serveur)*
- **Appelable**
  - *par d'autres procédures stockées*
  - *par des triggers*
  - *par du code externe (JAVA, C#, PHP)*
- **MySQL**  
*inclus à partir de la version 5.0*

# AVANTAGES DES PROCEDURES STOCKEES



- **Performantes**  
*compilées et stockées dans la base*  
*spécificité MySQL : compilation à la demande + mise en cache (par connexion)*
- **Diminue le trafic entre l'application et la base de données**  
*envoi uniquement d'un nom et de paramètres*
- **Réutilisables**  
*interface à la base utilisables par différentes applications*  
*pas de redéveloppement*
- **Sécurisées**  
*attribution du droit d'exécution de la procédure stockées*  
*sans avoir à donner les droits d'accès à la base sous-jacente*

# INCONVENIENTS DES PROCEDURES STOCKEES



- **Consommatrices en ressources**  
*mémoire (connexion)*  
*CPU ( en cas de nombreuses opérations logiques)*
- **Difficile d'intégrer des logiques métiers complexes**
- **Difficiles à débbugger**  
*manque d'environnement adéquate*
- **Difficiles à développer et maintenir**  
*développeur spécialisé*

# MA PREMIERE PROCEDURE STOCKEE

- **En ligne de commande**  
*dans la fenetre de commande, lancer la commande mysql*

```
mysql> use cdbase;  
Database changed  
mysql> DELIMITER $$  
mysql> CREATE PROCEDURE listeAlbum ()  
mysql> BEGIN  
mysql>     SELECT * FROM album ORDER BY titre;  
mysql> END $$  
Query OK, 0 rows affected (0.00 sec)  
mysql> DELIMITER ;  
mysql>
```

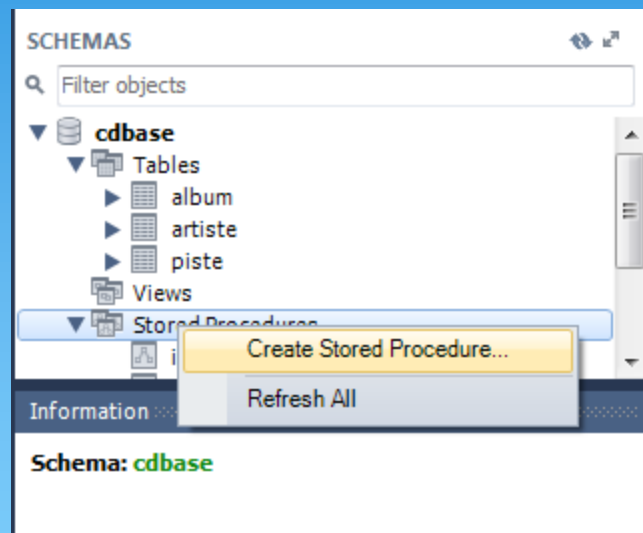
Change le délimiteur

Création de la procédure stockée (CREATE PROCEDURE)

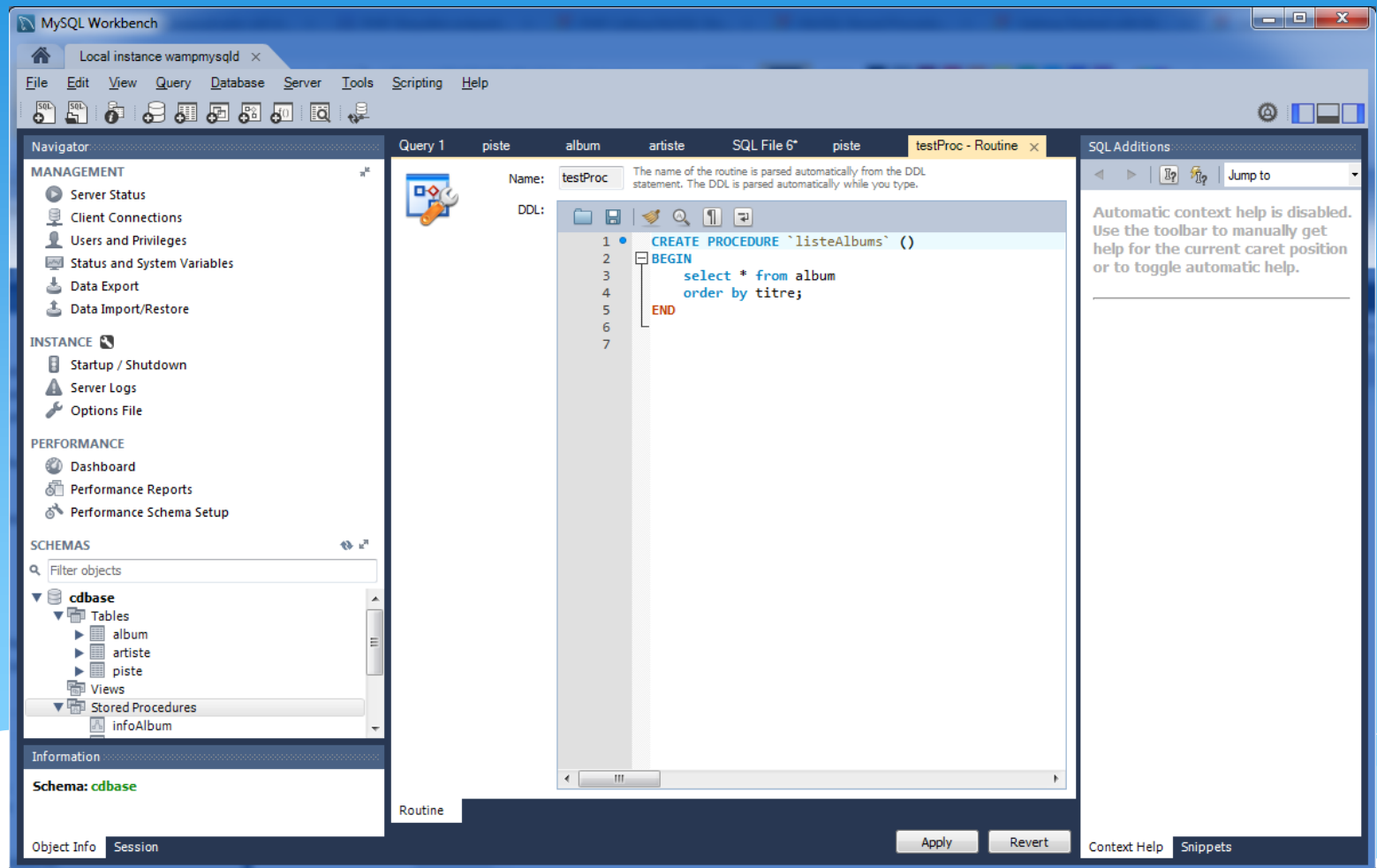
Nom de la procédure (suivi de parenthèse)

Corps de la procédure stockées  
(BEGIN ... END)

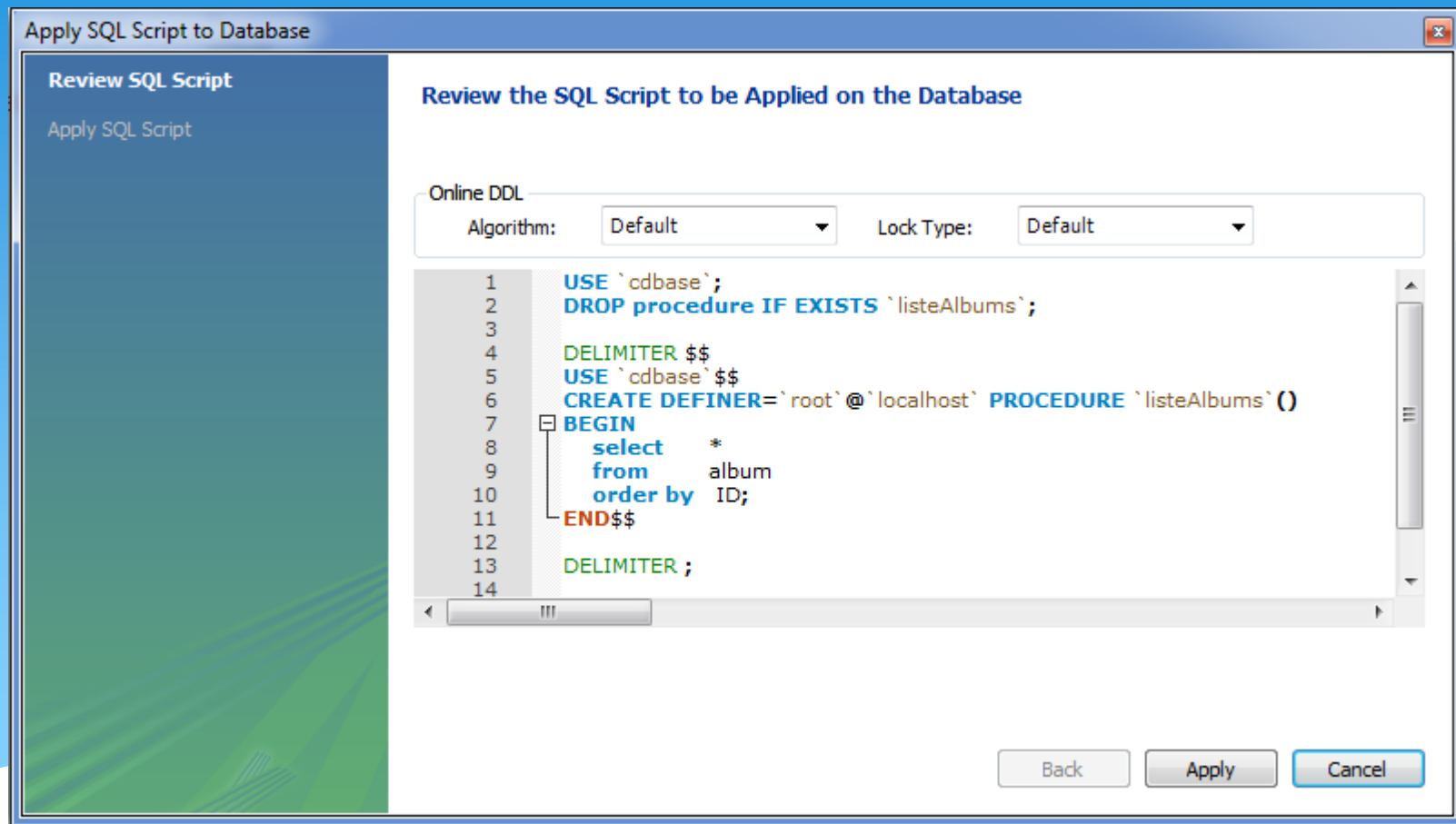
# VIA MySQL WorkBench (1)



# VIA MySQL WorkBench (2)

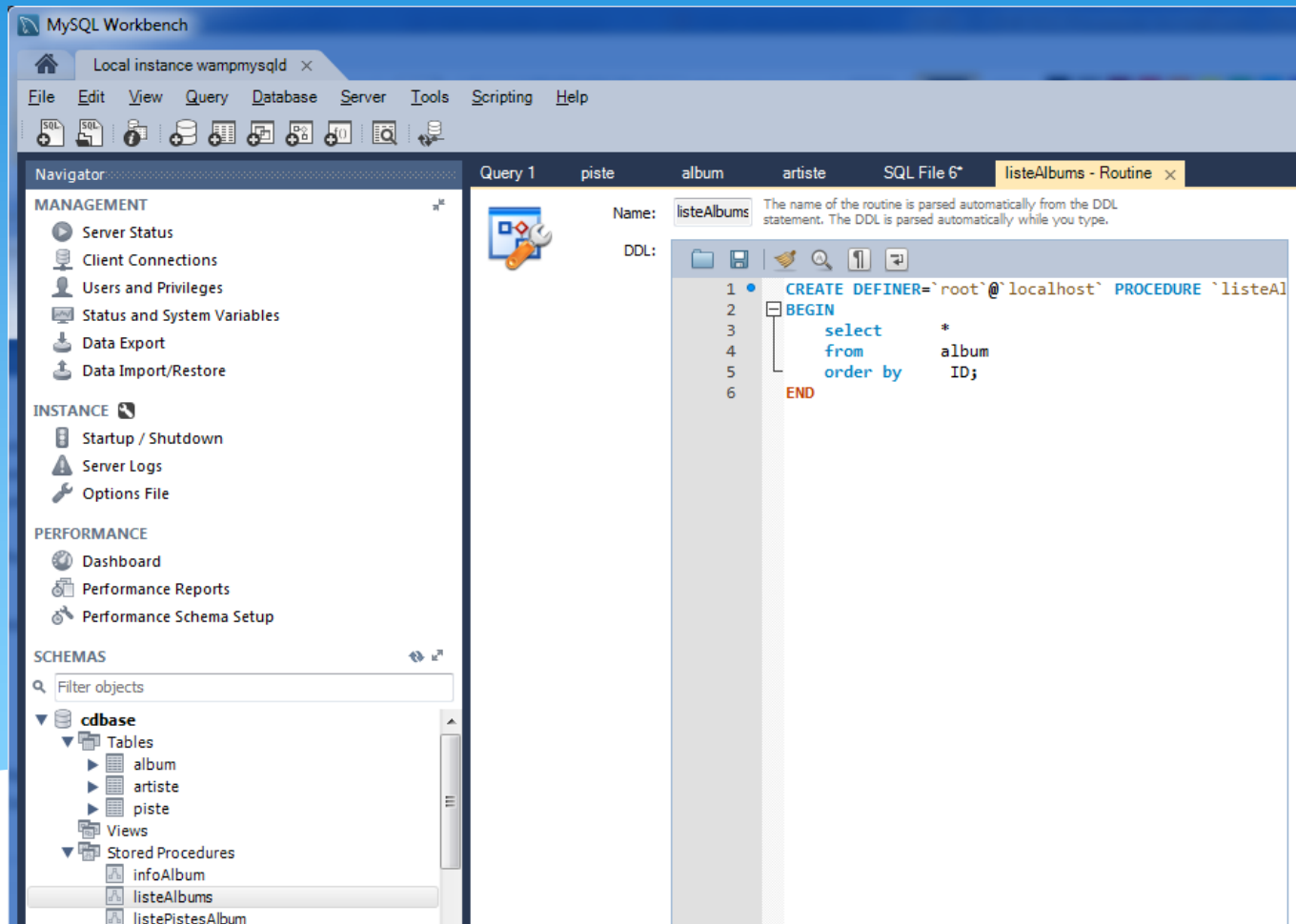


# VIA MySQL WorkBench (3)

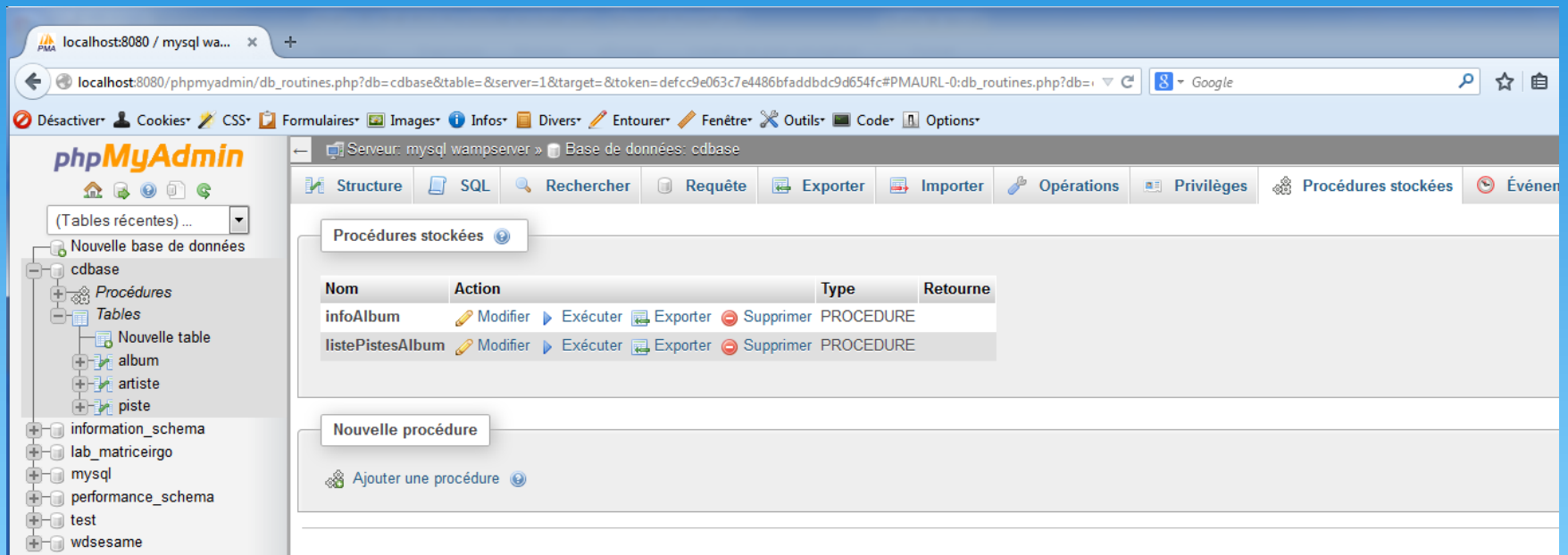




# VIA MySQL WorkBench (4)



# VIA PHPMYADMIN (1)



The screenshot shows the phpMyAdmin web interface in a browser window. The address bar indicates the URL is localhost:8080/phpmyadmin/db\_routines.php?db=cdbase&table=&server=1&target=&token=defcc9e063c7e4486bfaddbd9d654fc#PMAURL-0:db\_routines.php?db=... The interface is in French. The left sidebar shows the database structure with 'cdbase' selected. The main panel displays 'Procédures stockées' (Stored Procedures) for the 'cdbase' database. It contains a table with two rows: 'infoAlbum' and 'listePistesAlbum', both of type 'PROCEDURE'. Each row has actions for 'Modifier' (Modify), 'Exécuter' (Execute), 'Exporter' (Export), and 'Supprimer' (Delete). Below the table, there is a section for 'Nouvelle procédure' (New procedure) with a button to 'Ajouter une procédure' (Add a procedure).

Nom	Action	Type	Retourne
infoAlbum	Modifier Exécuter Exporter Supprimer	PROCEDURE	
listePistesAlbum	Modifier Exécuter Exporter Supprimer	PROCEDURE	

# VIA PHPMYADMIN (2)

Ajouter une procédure

Détails

Nom de la procédure

listeAlbum

Type

PROCEDURE

Paramètres

Direction

Nom

Type

Taille/Valeurs\*

Options

Ajouter un paramètre

Définition

```
1 select *
2 from album
3 order by titre;
```

Est déterministe

☐

Créateur

Type de sécurité

DEFINER

Accès aux données SQL










READS SQL DATA


Commentaire

Exécuter

Fermer













# VIA PHPMYADMIN (2)

 Structure
  SQL
  Rechercher
  Requête
  Exporter
  Importer
  Opérations
  Privilèges
  Procédures stockées

 La procédure 'listeAlbum' a été créée.

```
CREATE PROCEDURE `listeAlbum`() NOT DETERMINISTIC READS SQL DATA SQL SECURITY DEFINER select * from album order by titre;
```

Procédures stockées

Nom	Action	Type	Retourne
infoAlbum	 Modifier  Exécuter  Exporter  Supprimer	PROCEDURE	
listeAlbum	 Modifier  Exécuter  Exporter  Supprimer	PROCEDURE	
listePistesAlbum	 Modifier  Exécuter  Exporter  Supprimer	PROCEDURE	

# ADDEL (SQL)

- Syntaxe

```
<?php
// connexion
$db = new
```

```
$req = $db->
```

```
...
?>
```

Seveur: mysql wampserver » Base de données: cdbase

Structure SQL Rechercher Requête Exporter Importer

✓ Votre requête SQL a été exécutée avec succès  
1 ligne a été affectée par le dernier énoncé de la procédure

```
CALL `listeAlbum` ();
```

Résultats de l'exécution de la procédure `listeAlbum`

ID	dateParution	IDArtiste	jaquette	titre
1	1987-03-09 00:00:00	1	The_Joshua_Tree.png	The Joshua Tree

Procédures stockées

Nom	Action	Type	Retourne
infoAlbum	Modifier ▶ Exécuter Exporter Supprimer	PROCEDURE	
listeAlbum	Modifier ▶ Exécuter Exporter Supprimer	PROCEDURE	
listePistesAlbum	Modifier ▶ Exécuter Exporter Supprimer	PROCEDURE	

# APPEL (SQL)

- **Syntaxe SQL**

**call nomProcedure ();**

The screenshot shows a database management interface with a toolbar at the top containing icons for Structure, SQL, Rechercher, Requête, Exporter, and Importer. A green message box states: "✓ Votre requête SQL a été exécutée avec succès. 1 ligne a été affectée par le dernier énoncé de la procédure." Below this, the SQL command `CALL 'listeAlbums'();` is displayed. A section titled "Résultats de l'exécution de la procédure 'listeAlbums'" contains a table with the following data:

ID	dateParution	IDArtiste	jaquette	titre
1	1987-03-09 00:00:00	1	The_Joshua_Tree.png	The Joshua Tree

Below the results, a section titled "Procédures stockées" displays a table of stored procedures:

Nom	Action	Type	Retour
infoAlbum	Modifier Exécuter Exporter Supprimer	PROCEDURE	
listeAlbums	Modifier Exécuter Exporter Supprimer	PROCEDURE	
listePistesAlbum	Modifier Exécuter Exporter Supprimer	PROCEDURE	

# APPEL (PHP)

```
<?php
// connexion à la base de données
$db = new PDO('mysql:host=localhost; dbname=maBase', $login, $mdp);

$req = $db->prepare("CALL listeAlbums()");
$req->execute();
while ($ligne= $req-> fetch(PDO::FETCH_ASSOC)){
    echo $ligne["titre"] . "( date parution :") . $ligne["dateParution"]. "<br/>";
}

...
?>
```

# PARAMETRES

- **Déclaration**

*mode nom\_param type\_param (taille\_param) [, ...]*

- **Mode**

<i>IN</i>	<i>parametre d'entrée (par défaut)</i>
<i>OUT</i>	<i>paramètre de sortie</i>
<i>INOUT</i>	<i>parametre d'entree sortie</i>

- **Type, taille**

*cf type SQL (int, float, varchar, datetime, ...)*

```
CREATE PROCEDURE listePistesDeLAlbum (IN p_IDAlbum INT)  
SELECT *  
FROM piste  
WHERE IDAlbum = p_IDAlbum  
ORDER BY ID
```



# APPEL (SQL)

```
CALL ListePistesDeLAlbum(17);
```

# APPEL (PHP)

```
<?php
// connexion à la base de données
$db = new PDO('mysql:host=localhost; dbname=maBase', $login, $mdp);

$req = $db->prepare("CALL listePistesDeLAlbum (:IDAlbum)");
$req->bindParam(":IDAlbum", $_GET['ID']);
$req->execute();
while ($ligne= $req-> fetch(PDO::FETCH_ASSOC)){
    echo "$ligne['ID'] - $ligne['titre'] ($ligne['duree']) <br/>";
}

...
?>
```

# FONCTIONS

- **Procédures stockées renvoyant une valeur scalaire**  
*integer, float, varchar, datetime, ...*

```
CREATE DEFINER='root'@'localhost' FUNCTION `nbPistes2`(p_IDAlbum integer) RETURNS int(11)
BEGIN
    declare nb integer;

    select count(*) into nb
    from piste
    where IDAlbum = p_IDAlbum;
    RETURN nb;
END
```

# APPEL (SQL)

```
SELECT nbPistes (17) as 'total'
```

# APPEL (PHP)

```
<?php
// connexion à la base de données
$db = new PDO('mysql:host=localhost; dbname=maBase', $login, $mdp);

$req = $db->prepare("SELECT nbPistesAlbum (:IDAlbum)");
$req->bindParam(":IDAlbum", $_GET['ID']);
$req->execute();
$nb = $req->fetchColumn();
echo "cet album a $nb pistes <br/>";
?>
```

# CONCLUSION