

Documentation classe MySql

Créer par Filipe Gomes

Fichier: class.mysql.php

Version: 1.2

INTRODUCTION

Cette classe très pratique vous permettra de gagner du temps lorsque vous aurez besoin de lancer des requêtes SQL. En effet, celle-ci a été conçue pour lancer une requête SQL à vitesse grand « V ».

De plus, cette classe protège toutes les données des injections SQL, hormis certains paramètres ou variable mais vous le verrez dans cette petite documentation.

Variables, Fonctions Et Méthodes

Voici un tableau contenant les principales variables, fonctions et méthodes de cette classe.

VARIABLES	DESCRIPTION
\$GetErrorMysql	Stocke la dernière erreur générées par la classe
\$DernierID	Stocke le dernier identifiant d'une requête INSERT
\$AfficherMessException	Permet de gérer l'affichage des erreurs. Mettez true pour afficher les erreurs, autrement mettez false . La valeur par défaut est true .
\$CloseMysqlAuto	Permet de gérer la fermeture de votre connexion MySql. Mettez true si vous voulez que la connexion soit fermée à chaque requête, autrement mettez false . La valeur par défaut est false .
\$ApercuCreateTableSql	Si vous créer une table SQL avec la méthode MySqlBddCreateTable(). Vous pouvez obtenir l'aperçu juste en faisant appel à cette variable.
\$SqlHost	Nom du serveur MySql (en local c'est localhost)
\$SqlUser	Nom utilisateur de la base de données (en local mettez root)
\$SqlPass	Mot de passe d'accès à la base de données (en local, laissez vide)
\$SqlBdd	Nom de la base de données
\$TempsExecution	Variable contenant le temps d'exécution en secondes d'une requête SQL
METHODES	DESCRIPTION
MySqlOpen()	Ouvre la connexion SQL
MySqlClose()	Ferme la connexion SQL
MySqlInsert()	Permet d'insérer une nouvelle ligne dans la base de données.
MySqlCount()	Retourne le nombre de lignes qui correspond à un critère spécifié
MySqlUpdate()	Permet de mettre à jour des enregistrements existants dans une table
MySqlDelete()	Cette méthode permet supprimer des enregistrements dans une table
MySqlSelect()	Permet d'afficher les éléments de votre base de données.
MySqlSelectPerso()	Identique à MySqlSelect() mais vous pouvez personnaliser votre ligne SQL à 100%
MySqlQuery()	Pour des requêtes complexes cette méthode pourra servir
MySqlBddDropTable()	Permet de supprimer une table SQL de votre base de données
MySqlBddCreateTable()	Permet de créer une nouvelle table SQL dans votre base de données
FONCTIONS	DESCRIPTION
MySqlProtectVal()	Protège les données contre les injections SQL

Ouverture De La Classe

Avant toute chose vous devez ouvrir la classe :

Code PHP

```
// Ouverture de la base de données
require("class.mysql.php");
// instantiation de la classe
$bdd = new MySQLClass();
// CONNEXION A LA BASE DE DONNEES
// Paramètre 1 : Nom du serveur
// Paramètre 2 : Nom utilisateur
// Paramètre 3 : Mot de passe vers la base de données
// Paramètre 4 : Nom de la base de données
$bdd->MysqlOpen("localhost", "root", "", "ma_table_sql");
```

Vous pouvez aussi utiliser cette méthode:

Code PHP

```
// Ouverture de la base de données
require("class.mysql.php");
// instantiation de la classe
$bdd = new MySQLClass();
// CONNEXION A LA BASE DE DONNEES
$bdd->SqlHost = "localhost"; // Nom du serveur
$bdd->SqlUser = "root"; // Nom utilisateur
$bdd->SqlPass = ""; // Mot de passe vers la base de données
$bdd->SqlBdd = "test"; // Nom de la base de données
$bdd->MysqlOpen();
```

Les Clauses

Dans la plupart des méthodes comme MysqlSelect(), MysqlCount, MysqlUpdate... vous pouvez insérer des clauses (WHILE, LIMIT, ORDER BY, LEFT JOIN etc...). Pour vous faciliter la mise en place vous avez 2 possibilités de les utiliser. Par String (chaîne) ou en utilisant des Array (tableaux).

Il est important de savoir que toutes les valeurs qui passerons par les clauses ne seront pas protégés contre les injections SQL. Vous pouvez éventuellement utiliser la fonction MySQLProtectVal() disponible dans cette classe dans chacune de vos valeurs.

Pour nos exemples nous allons utiliser la méthode MysqlSelect() mais le fonctionnement est exactement le même avec les autres méthodes mais attention à l'emplacement du paramètre qui peut-être différent de celui-ci (voir plus bas)

Code PHP

```
// Insertion d'une clause WHERE au format String (chaîne)
$MysqlSelect("table", "*", "WHERE Colonel='Valeur 1' AND id=9");
// La même chose mais au format Array (tableau)
$MysqlSelect("table", "*", array(array("WHERE" => "Colonel='Valeur 1' AND id=9")));

// On peut aussi utiliser plusieurs clauses :
$MysqlSelect("table", "*", "WHERE Colonel='Valeur 1' ORDER BY id DESC LIMIT
1,9");

// Au format Array.
// Attention, le 1er paramètre doit-être déclarer qu'une seule fois.
// Si vous voulez continuer par exemple un WHERE il suffit simplement de faire ainsi:
// array("WHERE" => "Colonel='Val 1' AND Colonne2='Val 2'")
$MysqlSelect("table", "*", array(
    array("WHERE" => "Colonel='Valeur 1'"),
    array("ORDER" => "id DESC"),
    array("LIMIT" => "1,9")
));
// au lieu de taper à chaque fois WHERE, ORDER et autres dans le 1er paramètre du array
// j'ai également facilité la mise en place comme dans cet exemple qui est
// identique à l'exemple ci-dessus. Vous verrez un tableau récapitulatif plus bas.
$MysqlSelect("table", "*", array(
    array("W" => "Colonel='Valeur 1'"),
    array("O" => "id DESC"),
    array("L" => "1,9")
));
```

TABLEAU CONTENANT LES PARAMETRES VALIDENT POUR L'UTILISATION DES CLAUSES EN ARRAY

Clauses	Raccourcis pour le 1er paramètre				
	Voici les raccourcis qui peuvent être utilisés dans les premiers paramètres de chaque Array pour les clauses				
WHERE LIKE	LIKE	WL	WHERE LIKE	WHERELIKE	WHERE_LIKE
LEFT JOIN	LEFT	LJ	LEFT JOIN	LEFTJOIN	LEFT_JOIN
RIGHT JOIN	RIGHT	RJ	RIGHT JOIN	RIGHTJOIN	RIGHT_JOIN
ORDER BY	ORDER	O	ORDER BY	ORDERBY	ORDER_BY
GROUP BY	GROUP	G	GROUP BY	GROUPBY	GROUP_BY
HAVING	HAVING	H			
WHERE	WHERE	W			
LIMIT	LIMIT	L			

MysqlSelect()

La méthode MysqlSelect() vous permet d'afficher des éléments de votre base de données. Son mode de fonctionnement est identique à la requête SELECT habituel mais en plus court et plus rapide.

Toutes les valeurs qui passeront par \$Param2 et \$Param3 ne seront pas protégés contre les injections SQL.

PARAMETRES DE LA METHODE**MySqlSelect(\$Param1, \$Param2, \$Param3, \$Param4);**

\$Param1 (string)	Nom de la table SQL de votre base de données
\$Param2 (string)	Colonnes à sélectionner. Par défaut c'est * (toutes les colonnes) <i>(facultatif)</i>
\$Param3 (array/string)	Clauses à intégrer (WHERE, ORDER BY, LIMIT etc...) au format Array ou String <i>(facultatif)</i>
\$Param4 (string)	Pour ne pas utiliser la boucle While, mettez false , autrement mettez true . La boucle While est utilisée pour afficher plusieurs lignes de votre base de données mais parfois elle n'est pas nécessaire. Par défaut c'est true . <i>(facultatif)</i>

Il existe plusieurs façons d'utiliser cette méthode dont voici quelques exemples :

Code PHP

Utiliser MySqlSelect avec \$Param4 à FALSE (sans utiliser de boucle WHILE)

```
// Ouverture de la requête
$ligne = $bdd->MySqlSelect('ma_table_sql', 'id,Colonne3', 'WHERE id=9', false);
// Affichage
echo '<div>Mon ID est '.$ligne['id'].' et ma colonne ' .
$ligne['Colonne3'].'</div>';
```

Code PHP

Utiliser MySqlSelect avec \$Param4 à TRUE (utiliser la boucle WHILE) exemple 1

```
// Ouverture de la requête
$req = $bdd->MySqlSelect('ma_table_sql', '*', 'ORDER BY id DESC');
// Affichage
for ($i=0;$i<count($req);$i++)
{
    $ligne = $req[$i];
    echo '
<div>'.$ligne['id'].'</div>
<div>'.$ligne['Colonne2'].'</div>
<div>'.$ligne['Colonne3'].'</div>';
}
```

Code PHP

Utiliser MySqlSelect avec \$Param4 à TRUE (utiliser la boucle WHILE) exemple 2

```
// Ouverture de la requête
$req = $bdd->MySqlSelect('ma_table_sql', '*', 'ORDER BY id DESC');
// Affichage
for ($i=0;$i<count($req);$i++)
{
    echo '
<div>'.$req[$i]['id'].'</div>
<div>'.$req[$i]['Colonne2'].'</div>
<div>'.$req[$i]['Colonne3'].'</div>';
}
```

Code PHP**Utiliser DISTINCT avec MySqlSelect**

```
// Ouverture de la requête
$req=$bdd->MySqlSelect('ma_table_sql', 'DISTINCT Colonne2', 'ORDER BY id DESC');
// Affichage
for ($i=0;$i<count($req);$i++)
{
    echo '
    <div>'.$req[$i]['id'].'</div>
    <div>'.$req[$i]['Colonne2'].'</div>
    <div>'.$req[$i]['Colonne3'].'</div>';
}
```

MySqlSelectPerso()

La méthode `MySqlSelectPerso()` fonctionne exactement comme la méthode `MySqlSelect()` mais vous pouvez personnaliser la ligne SQL à 100%. Attention, les tableaux `array()` ne sont pas disponibles dans cette méthode.

Toutes les valeurs qui passeront par `$Param1` ne seront pas protégés automatiquement contre les injections SQL.

PARAMETRES DE LA METHODE

MySqlSelectPerso(\$Param1, \$Param2);

<code>\$Param1</code> (string)	Requête SQL personnalisée, la ligne entière doit être tapée
<code>\$Param2</code> (string)	Pour ne pas utiliser la boucle <code>While</code> , mettez <code>false</code> , autrement mettez <code>true</code> . La boucle <code>While</code> est utilisée pour afficher plusieurs lignes de votre base de données mais parfois elle n'est pas nécessaire. Par défaut c'est <code>true</code> . <i>(facultatif)</i>

Code PHP**Exemple d'utilisation de MySqlSelectPerso**

```
// Ouverture de la requête
$req = $bdd->MySqlSelectPerso("SELECT * FROM table WHERE Colonne1='Vall' ORDER BY id");
// Affichage
for ($i=0;$i<count($req);$i++)
{
    $ligne = $req[$i];
    echo '
    <div>'.$ligne['id'].'</div>
    <div>'.$ligne['Colonne2'].'</div>
    <div>'.$ligne['Colonne3'].'</div>';
}
```

MysqlDelete()

La méthode MysqlDelete() vous permet de supprimer une ligne de votre table SQL.

Toutes les valeurs qui passeront par \$Param2 ne seront pas protégés contre les injections SQL. Vous pouvez éventuellement utiliser la fonction MySqlProtectVal() disponible dans cette classe dans chacune de vos valeurs.

PARAMETRES DE LA METHODE

MysqlDelete(\$Param1, \$Param2);

\$Param1 (string)	Nom de la table de votre base de données
\$Param2 (array/string)	Clauses à intégrer (WHERE, ORDER BY, LIMIT etc...) au format Array ou String (<i>facultatif</i>)

Code PHP

Exemple d'utilisation de MysqlDelete

```
// Suppression d'une ligne SQL
$bdd->MysqlDelete('ma_table_sql', 'WHERE id=1 AND id=2');
```

Code PHP

Exemple d'utilisation de MysqlDelete avec array

```
// La même chose mais au format Array (tableau)
$bdd->MysqlDelete("ma_table_sql", array(
    array("WHERE" => "id=1 AND id=2")
));
```

MysqlUpdate()

La méthode MysqlUpdate() vous permet de mettre à jour une ou plusieurs lignes de votre table SQL.

Toutes les valeurs qui passeront par \$Param3 ne seront pas protégés contre les injections SQL. Vous pouvez éventuellement utiliser la fonction MySqlProtectVal() disponible dans cette classe dans chacune de vos valeurs.

PARAMETRES DE LA METHODE

MysqlUpdate(\$Param1, \$Param2, \$Param3);

\$Param1 (string)	Nom de la table SQL de votre base de données
\$Param2 (array)	Nom des colonnes et valeurs à mettre à jour (valeurs protégées contre les injections SQL)
\$Param3 (array/string)	Clauses à intégrer (WHERE, ORDER BY, LIMIT etc...) au format Array ou String (<i>facultatif</i>)

Code PHP**Exemple 1**

```
// Mise à jour d'une ligne SQL
$dbdd->MysqlUpdate( "ma_table_sql", array(
    "Colonne1" => "Nouvelle valeur",
    "Colonne2" => "Nouvelle valeur"
),
"WHERE id=1");
```

Code PHP**Exemple 2**

```
// Mise à jour d'une ligne SQL
$dbdd->MysqlUpdate( 'ma_table_sql',
    array(
        "Colonne1" => "Nouvelle valeur",
        "Colonne2" => "Nouvelle valeur"
    ),
    array(
        array( "WHERE" => "id=1" )
    )
);
```

MysqlCount()

La méthode MysqlCount() retourne le nombre de lignes qui correspond à un critère spécifique.

Toutes les valeurs qui passeront par \$Param2 et \$Param3 ne seront pas protégés contre les injections SQL. Vous pouvez éventuellement utiliser la fonction MySqlProtectVal() disponible dans cette classe dans chacune de vos valeurs.

PARAMETRES DE LA METHODE

MysqlCount(\$Param1, \$Param2, \$Param3);

\$Param1 (string)	Nom de la table SQL de votre base de données
\$Param2 (string)	Nom de la colonne à compter, par défaut c'est * (tout) <i>(facultatif)</i>
\$Param3 (array/string)	Clauses à intégrer (WHERE, ORDER BY, LIMIT etc...) au format Array ou String <i>(facultatif)</i>

Code PHP**Exemple 1**

```
$dbdd->MysqlCount( "ma_table_sql", "id", "WHERE Colonne1='Valeur' AND
Colonne2='Valeur' " );
```


Code PHP**Exemple 2**

```
$bdd->MysqlCount("ma_table_sql", "*", array(
    array("WHERE" => "Colonne1='Valeur' AND Colonne2='Valeur'")
));
```

Code PHP**Exemple 3**

```
// Dans cet exemple tout de "ma_table_sql" sera compté
$bdd->MysqlCount("ma_table_sql");
```

MysqlInsert()

La méthode MysqlInsert() permet d'insérer des nouveaux éléments dans la base de données.

PARAMETRES DE LA METHODE

MysqlInsert(\$Param1, \$Param2);

\$Param1 (string)	Nom de la table SQL de votre base de données
\$Param2 (array)	Elements à insérer dans la base de données (<i>valeurs protégées contre les injections SQL</i>)

Code PHP**Exemple**

```
$bdd->MysqlInsert("ma_table_sql", array(
    "Colonne1" => "Valeur à insérer",
    "Colonne2" => "Valeur à insérer"
));
```

MysqlQuery()

Cette méthode est utile pour des requêtes SQL plus complexes

PARAMETRES DE LA METHODE

MysqlQuery(\$Param);

\$Param (string)	Requête SQL complète
------------------	----------------------

Code PHP**Exemple**

```
// pour l'exemple je fait simple, je ne cherche pas les complications
$req=$bdd->MysqlQuery("SELECT * FROM matable WHERE id=1");
while($ligne=mysql_fetch_array($req))
{
    echo "<div>".$ligne['ma_colonne']."</div>";
}
mysql_close(); // si vous voulez fermer votre connexion
```

MysqlClose()

La méthode MysqlClose() permet de fermer votre connexion MySQL.

Code PHP**Exemple**

```
// Fermeture de la base de données
$bdd->MysqlClose();
```

MysqlBddCreateTable()

Cette méthode vous permet de créer une table SQL.

Plus bas dans cette section, vous avez des tableaux avec les paramètres des Index et Types.

PARAMETRES DE LA METHODE

MysqlBddCreateTable(\$Param1, \$Param2, \$Param3, \$Param4);

\$Param1 (string)	Nom de la table SQL que vous souhaitez créer
\$Param2 (array)	Éléments à insérer dans votre table. Les paramètres obligatoires sont : NOM, TYPE et TAILLE
\$Param3 (string)	Moteur de la table. Par défaut c'est InnoDB. <i>(facultatif)</i>
\$Param4 (string)	Charset par défaut, la valeur par défaut est latin1. <i>(facultatif)</i>

Code PHP**Exemple**

```
// 1ere colonne
$colonne1 = array(
    "NOM" => "id", // Nom de la colonne (obligatoire)
    "TYPE" => "INT", // Type de colonne (obligatoire)
    "TAILLE" => "11", // Taille de la colonne (obligatoire)
    "VALEUR_DEFAULT" => "", // Valeur par défaut
```

```

"NULL" => "1", // NULL (laissez vide le cas contraire)
"AUTO_INCREMENT" => "1", // Colonne auto incrémentée (laissez vide le cas contraire)
"INDEX" => "PRIMARY", // Index de la table (un seul index par table)
"COMMENTAIRES" => "Com" // Commentaire
);
// 2e colonne
$colonne2 = array("NOM"=>"nom_colonne2", "TYPE"=>"VARCHAR", "TAILLE"=>"255",
"VALEUR_DEFAUT"=>"Ma valeur par défaut");
// 3e colonne
$colonne3 = array("NOM"=>"nom_colonne3", "TYPE"=>"TEXT", "TAILLE"=>"");
// Création de la table
$mysqlBddCreateTable('ma_nouvelle_table', array($colonne1, $colonne2,
$colonne3));

```

TABLEAU DES TYPES POUR CREER UNE TABLE SQL

Présentation des types numériques

TINYINT	Un très petit entier. L'intervalle de validité pour les entiers signés est de -128 à 127. L'intervalle de validité pour les entiers non-signés est 0 à 255.
BIT, BOOL, BOOLEAN	Éléments à insérer dans votre table. Les paramètres obligatoires sont : NOM, TYPE et TAILLE. Ce sont des synonymes de TINYINT(1). Le synonyme BOOLEAN a été ajouté en version 4.1.0. Un type booléen complet, qui sera introduit pour être en accord avec la norme SQL-99.
SMALLINT	Un petit entier. L'intervalle de validité pour les entiers signés est de -32768 à 32767. L'intervalle de validité pour les entiers non-signés est 0 à 65535.
MEDIUMINT	Un entier. L'intervalle de validité pour les entiers signés est de -8388608 à 8388607. L'intervalle de validité pour les entiers non-signés est 0 à 16777215.
INT	Un grand entier. L'intervalle de validité pour les entiers signés est de -2147483648 à 2147483647. L'intervalle de validité pour les entiers non-signés est 0 à 4294967295.
BIGINT	Un très grand entier. L'intervalle de validité pour les entiers signés est de -9223372036854775808 à 9223372036854775807. L'intervalle de validité pour les entiers non-signés est 0 à 18446744073709551615.
FLOAT	Un petit nombre à virgule flottante, en précision simple. Les valeurs possibles vont de -3.402823466E+38 à -1.175494351E-38, 0, et 1.175494351E-38 à 3.402823466E+38.
DOUBLE	Un nombre à virgule flottante, en précision double. Les valeurs possibles vont de -1.7976931348623157E+308 à -2.2250738585072014E-308, 0, et 2.2250738585072014E-308 à 1.7976931348623157E+308.
DECIMAL	Un nombre à virgule flottante littéral. Il se comporte comme une colonne de type CHAR: ``littéral'' (``unpacked'') signifie que le nombre est stocké sous forme de chaîne : chaque caractère représente un chiffre. La virgule décimale et le signe moins '-' des nombres négatifs ne sont pas comptés dans M (mais de l'espace leur est réservé).

Présentation des types chaînes

CHAR	Une chaîne de caractère de taille fixe, toujours complétée à droite par des espaces pour remplir l'espace de stockage. L'intervalle de la taille va de 0 à 255 (1 à 255 pour les versions antérieures à la version 3.23).
VARCHAR	Une chaîne de taille dynamique. L'intervalle de la taille va de 0 à 255 caractères (1 à 255 avant MySQL 4.0.2).
TINYTEXT	Une colonne TINYTEXT peut contenir au maximum 255 (2 ⁸ - 1) caractères.

TINYBLOB	Identique à TINYTEXT
TEXT	Une colonne TEXT peut contenir au maximum 65535 (2 ¹⁶ – 1) caractères.
BLOB	Identique à TEXT
MEDIUMBLOB	Une colonne MEDIUMBLOB peut contenir au maximum 16777215 (2 ²⁴ – 1) caractères.
MEDIUMTEXT	Une colonne MEDIUMTEXT peut contenir au maximum 16777215 (2 ²⁴ – 1) caractères.
LONGTEXT	Une colonne LONGTEXT peut contenir au maximum 4294967295 ou 4 Go (2 ³² – 1) caractères.
LOBLOB	Identique à LONGTEXT
Présentation des types date et heure	
DATE	Une date. L'intervalle supporté va de '1000-01-01' à '9999-12-31'. MySQL affiche les valeurs de type DATE au format 'YYYY-MM-DD', mais vous permet d'assigner des valeurs DATE en utilisant plusieurs formats de chaînes et nombres.
DATETIME	Une combinaison de date et heure. L'intervalle de validité va de '1000-01-01 00:00:00' à '9999-12-31 23:59:59'. MySQL affiche les valeurs de type DATE au format 'YYYY-MM-DD HH:MM:SS', mais vous permet d'assigner des valeurs DATE en utilisant plusieurs formats de chaînes et nombres.
TIMESTAMP	L'intervalle de validité va de '1970-01-01 00:00:00' à quelque part durant l'année 2037. En MySQL 4.0 et plus récent, les valeurs TIMESTAMP sont affichées au format YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD ou YYMMDD, suivant que la taille est 14 (ou absente), 12, 8 ou 6, respectivement, mais vous permet d'assigner des valeurs aux colonnes TIMESTAMP en utilisant des nombres ou des chaînes.
TIME	Une heure. L'intervalle va de '-838:59:59' à '838:59:59'. MySQL affiche les valeurs TIME au format 'HH:MM:SS', mais vous permet d'assigner des valeurs TIME en utilisant des nombres ou des chaînes.
YEAR	Une année, au format 2 ou 4 chiffres (par défaut, c'est 4 chiffres). Les valeurs possibles vont de 1901 à 2155 plus 0000 pour le format à 4 chiffres, et de 1970 à 2069 si vous utilisez le format à 2 chiffres. MySQL affiche les valeurs YEAR au format YYYY mais vous permet d'assigner des valeurs en utilisant des nombres ou des chaînes. Le type YEAR n'est pas disponible avant la version 3.22.

TABLEAU DES INDEX

PRIMAIRE	Une clé primaire (PRIMARY KEY) est strictement unique, les NULL ne sont pas autorisés.
UNIQUE	Un index de type UNIQUE est comparable à une clé primaire, mis à part pour les valeurs NULL puisque celles-ci sont autorisées (et potentiellement en plusieurs occurrences).
INDEX	Un index de type INDEX ou KEY (c'est un alias) signifie simplement que l'on souhaite indexer une colonne susceptible de contenir des doublons.
FULLTEXT	Les index FULLTEXT ne peuvent indexer que des colonnes VARCHAR ou TEXT, et seulement dans les tables MyISAM. Les index FULLTEXT sont disponibles dans les versions 3.23.23 et supérieures de MySQL.

MysqlBddDropTable()

Supprime une table de votre base de données (ne pas confondre avec la clause DELETE).

ATTENTION: CETTE ACTION SUPPRIME VOTRE TABLE DE LA BASE DE DONNEES MAIS AUSSI TOUTES LES DONNEES PRESENTES ET SERONT PERDUES DEFFINITIVEMENT.

PARAMETRES DE LA METHODE

MysqlBddDropTable(\$Param1);

\$Param1 (string)	Nom de la table SQL que vous souhaitez supprimer
-------------------	--

Code PHP

Exemple

```
// Suppression de la table SQL ainsi que toutes les données  
$bdd->MysqlBddDropTable("table_a_supprimer");
```