UPDATE : modifier les données d'une table MySQL

Date de dernière mise à jour : 05/02/2014 à 13:51

UPDATE table : Modifier des données dans une table MySQL

Dans l'article précédent, nous avons vu comment insérer des données dans une table avec MySQL. Voyons maintenant comment mettre à jour ces données.

Pour mettre à jour des données en MySQL (et plus généralement en SQL), vous devrez utiliser la commande UPDATE. Celle-ci permet de mettre à jour des enregistrements. Bien entendu, vous allez pouvoir spécifier quels enregistrements vous souhaiterez mettre à jour, au moyen de conditions.

Nous reprendrons notre exemple vu dans l'article précédent de la table des membres dont voici (pour rappel) la structure :

Afficher | Structure | SQL | Rechercher | Insérer | Expo

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	id	int(11)			Non	*Aucune*	AUTO_INCREMENT
2	pseudo	varchar(30)	latin1_swedish_ci		Non	*Aucune*	
3	passe	varchar(40)	latin1_swedish_ci		Non	*Aucune*	
4	email	varchar(100)	latin1_swedish_ci		Non	*Aucune*	

Si vous souhaitez reproduire cette table, vous pouvez exécuter le code SQL suivant (un outil tel que PHPMyAdmin est recommandé car plus pratique) :

```
CREATE
TABLE
IF
NOT
EXISTS
 `membres` (
  `id
` int
(
11
)
NOT
NULL
 AUTO_INCREMENT,
  `pseudo` varchar
(
30
)
NOT
NULL
,
  `passe` varchar
(
40
)
```

NOT
NULL
,
`email` varchar
(
100
)
NOT
NULL
,
PRIMARY
KEY
(
`id
`)
)
ENGINE=
InnoDB DEFAULT
CHARSET=
latin1;

Voici les quelques données qui figurent dans notre table **membres** :

Vous pouvez les insérer en exécutant les requêtes suivantes :

INSERT
INTO
`
membres
`
(
`
id
`
,
`
pseudo
`
,
`
passe
`
,
`
email

```
`
)
VALUES
(
1
,
'Pierre'
,
'fe1fb20ff84babba7e6ea3dcc4d1ad541d52a675'
,
'pierre@dupont.fr'
)
,
(
2
,
'Dupont'
,
'ff019a5748a52b5641624af88a54a2f0e46a9fb5'
,
'dupont@etdupont.fr'
)
;
```

Syntaxe de la requête UPDATE

La syntaxe des requêtes UPDATE est relativement simple. En premier, vous devez toujours spécifier le mot UPDATE, suivi du nom de votre table. Dans notre cas, cela donnera **UPDATE `membres`**. Vous pouvez omettre les caractères d'échappement ` si votre table ne contient pas de caractères spéciaux, ce qui est généralement le cas.

Ensuite, vous devez spécifier quels sont les champs que vous souhaiteriez modifier. Cela se fait avec le mot clé **SET** lui-même suivi des noms de champs (séparés par des virgules) et des valeurs à remplacer. Imaginions que nous souhaitions modifier le pseudo du membre Pierre (identifié par le numéro 1) et que nous souhaitions modifier le champ **email** pour y mettre la valeur **contact@pierre.fr**. La requête à effectuer est la suivante :

```
UPDATE
`
membres
`
SET
`
email
`
=
"p@pierre.fr"
WHERE
`
id
`
=
1
;
```

Vous pouvez constater la présence du mot-clé **WHERE**. Ce mot-clé est important puisque c'est lui qui va dire à MySQL de ne modifier que le membre dont le champ **id** vaut 1. Vous constatez par là l'importance d'avoir pour chaque enregistrement une **clé unique**, représentée par un champ dont la

valeur est toujours différente pour chacune des lignes. C'est le cas de notre champ **id** qui est auto-incrémenté.

Modifier plusieurs champs simultanément

Vous n'êtes pas obligé(e) d'effectuer plusieurs requêtes en modifiant un champ à la fois. Imaginions que nous souhaitions modifier le champ **email** et le champ **pseudo** du membre dont l'identifiant unique est **1**. Pour ce faire, la requête aurait pris la forme suivante :

```
UPDATE
`
membres
`
SET
`
email
`
=
"jacques@monfai.fr"
,
`
pseudo
`
=
"Jacques"
WHERE
`
id
`
=
1
;
```

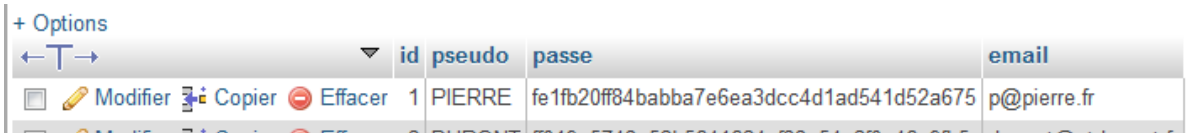
Notre membre dont l'identifiant est **1** porte désormais le pseudo **Jacques** et l'email **jacques@monfai.fr**.

Utilisation des fonctions dans les requêtes UPDATE

L'utilisation des fonctions se fait de la même façon que dans les requêtes INSERT, vues précédemment. Nous allons mettre tous nos pseudonymes en majuscules. La fonction MySQL permettant de faire cela s'appelle **UPPER()** et prend en paramètre la chaîne de caractères souhaitée.

```
UPDATE
`
membres
`
SET
`
pseudo
`
=
UPPER
(
`
pseudo
`
)
;
```

Remarquez que nos pseudos sont désormais tous en majuscules :



L'absence de clause **WHERE** indique que vous allez mettre à jour tous les enregistrements de la table.

Attention à ne pas confondre le caractère d'échappement ` avec une guillemet simple, qui n'a pas du tout la même signification ! Essayez de remplacer ce caractère par une guillemet simple dans la fonction UPPER() et voyez le résultat ... UPDATE `membres` SET `pseudo`=UPPER('pseudo');

Horreur ! tous vos membres s'appellent désormais **PSEUDO** !