

Modification d'enregistrements dans une base de données



Pour le moment, nous avons vu comment lire le contenu d'une base de données, et y insérer de nouveaux enregistrements. Par exemple, à chaque nouvelle personne qui s'inscrit, une nouvelle ligne est insérée dans la table "participants".

Mais comment faire lorsque l'on souhaite modifier un enregistrement, sans le supprimer, ni créer une nouvelle ligne? La solution, c'est l'utilisation de la requête UPDATE. Nous en profiterons pour voir comment récupérer une variable définie par l'utilisateur sans passer par un formulaire.

Préparation des fichiers

Nous n'allons travailler, cette fois, que sur la table "recompenses". En effet, l'exemple est facile à appliquer par la suite aux autres fichiers... En fait, voici ce qu'il se passera depuis la page d'accueil : pour ce qui est des lots mis en jeu, l'utilisateur pourra cliquer sur un lien "modifier/supprimer". Ce lien le conduira à une page qui listera tous les lots, et lui donnera la possibilité, lot par lot, soit de modifier, soit de supprimer l'enregistrement.

Pour le moment, nous avons donc trois pages à préparer :

Fichiers	rôle
intervention_lots.asp	listier tous les lots, et proposer un lien pour modifier le détail de chacun
modifier_lot.asp	modifier le lot qui a été sélectionné dans la page précédente
traitement_modif.asp	valider la modification

L'utilisateur a d'abord accès à **intervention_lots.asp**. A partir de là, il clique sur un lot et accède à **modifier_lot.asp**, qui lui permet de modifier le détail de chaque champ de ce lot.

Le fichier "intervention_lots.asp"

Ce fichier doit permettre de visualiser rapidement le contenu de la table, et de proposer un lien ("modifier") en face de chaque enregistrement.

Pour ce qui est de lire le contenu de la table, il suffit d'écrire un **SELECT** comme dans le chapitre précédent :

Création d'une **connexion** à une base, et d'un **recordset** :

```
DSN_BASE = "DBQ=" & Server.MapPath("marathon2000.mdb") & ";Driver={Microsoft Access Driver (*.mdb)};DriverId=25"
Set ConnTable = Server.CreateObject("ADODB.Connection")
ConnTable.Open DSN_BASE
Set RsTable = Server.CreateObject("ADODB.Recordset")
```

 **ouverture** du recordset :

```
RsTable.Open "Select * from recompenses" , ConnTable
```

ici encore, on sélectionne tous les champs de la table "récompenses".

 **gestion de l'une erreur** : EOF=BOF

il faut toujours prévoir l'erreur susceptible d'être annoncée si la table est vide :

```
if RsTable.eof = true then
<br>aucune récompense ne figure encore dans la
table!
```

- ce qui signifie : "si le recordset arrive pas à la fin du fichier, alors affiche le texte que voilà" ;
- ce qui sous-entend : "sinon va directement au **ELSE** qui suit".


 **réalisation d'une boucle** pour afficher tous les enregistrements trouvés :

il faut ici prévoir une cellule supplémentaire pour proposer de modifier l'enregistrement concerné :


```
<table>
<%'-----définition de la ligne des titres de colonnes-----%>
<tr>
<td>lot</td>
<td>sponsor</td>
<td>description</td>
<td>position</td>
<td>modifier/supprimer</td>
</tr>
<%'-----affichage des résultats, une ligne par enregistrement-----%>
<%RsTable.MoveFirst
DO WHILE not RsTable.eof%>
<tr>
<td><B><%=RsTable("lot")%></B></td>
<td><%=RsTable("sponsor")%></td>
<td><%=RsTable("description")%></td>
<td><%=RsTable("place_lot")%></td>
<td>
<a href="modifier_lot.asp?reference=<%=RsTable("ref_lot")%>">modifier</a>
</td>
</tr>
<%RsTable.MoveNext
loop%>
<%'----- Fin de la boucle, et de la ligne-----%>
</table>
```

La ligne surlignée est intéressante. Vous voyez que si l'on clique sur le mot "modifier", on est redirigé vers la page "modifier_lot.asp" ; le "?" signifie que vous y portez une variable, qui s'appelle "reference", et qui a pour valeur la **référence du lot** que vous vouliez modifier!

Nous le verrons plus loin, cette variable se récupère dans modifier_lot.asp avec un Request.QueryString ("reference").

 **fermeture** de la gestion de l'erreur

```
end if
```


 **fermeture** de la connexion et du recordset, **vidage** de leur contenu

```
RsTable.Close
Set RsTable= Nothing
ConnTable.Close
Set ConnTable = Nothing
```

Le fichier "modifier_lot.asp" :

Ce fichier va permettre de modifier l'enregistrement sélectionné. Plus précisément il nous faut :

- récupérer la référence du lot à modifier ;
- afficher le contenu de la table pour cet enregistrement dans un formulaire, qui permette à l'utilisateur de faire ses modifications.

 **Récupération** de la variable "référence" :


```
RefLot=Request.QueryString("reference")
```

Ici, nous créons une variable RefLot qui stocke cette donnée.

Attention!

Toutes les variables récupérées par un **REQUEST** sont de type texte! Or la référence du lot étant numérique, il nous faut faire une conversion avec **CINT** :

```
RefLot=CINT(RefLot)
```

 **Déclaration** de la base, création d'une **connexion** et d'un **recordset** :

Puisque nous allons récupérer des informations contenues dans la table, il faut à nouveau la déclarer, et créer connexion et recordset...

```

DSN_BASE = "DBQ=" & Server.MapPath("marathon2000.mdb") & ";Driver={Microsoft Access Driver (*.mdb)};DriverId=25"

Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open DSN_BASE
Set rs = Server.CreateObject("ADODB.Recordset")

```

Ouverture du recordset :

Il faut sélectionner tout ce qui concerne l'enregistrement qui porte comme référence RefLot :

```
rs.Open "Select * from recompenses WHERE ref_lot=" & RefLot , Conn
```

Affichage, sous la forme d'un **formulaire**, du contenu de chaque champ :

Ici c'est simple, il suffit de reprendre le formulaire qui nous permettait de créer un nouvel enregistrement! La différence est ici : tous les champs de ce formulaire auront comme valeur par défaut la valeur qui existe déjà dans la table...

exemple :

```
<INPUT type="text" name="nomlot" value="<%=rs("lot")%>" size="20">
```

(champ texte de taille 20 caractères, la variable "nomlot" contiendra la valeur saisie)

- les données du formulaires sont envoyées à la page "traitement_modif.asp". Nous utiliserons la méthode POST.

```
<form method="POST" name="ModifLot" action="traitement_modif.asp">
```

- il faut y rajouter un champ caché (c'est à dire qui ne sera pas apparent dans la page, mais qui sera tout de même renvoyé par le formulaire) pour récupérer la référence du lot à modifier :

```
<INPUT type="hidden" name="RefLot" value="<%=RefLot%>" size="20">
```

fermeture de la connexion et du recordset, **vidage** de leur contenu

```

rs.Close
Set rs= Nothing
Conn.Close
Set Conn = Nothing

```

Le fichier "traitement_modif.asp" :

C'est le fichier qui récupère les informations que l'utilisateur a saisi dans chaque champ du formulaire "ModifLot". Comme le formulaire a été envoyé avec la méthode POST, les informations sont récupérées

avec Request.Form("Nom_du_champ") - voir la [FAQ pour plus de précisions](#) sur ce sujet.

On définit des variables, pour recevoir le contenu de ces informations :

```
lot_bis = Request.Form("nomlot")
```

On ouvre la base, la connexion, un recordset :

```
DSN_BASE = "DBQ=" & Server.MapPath("marathon2000.mdb") & ";Driver={Microsoft Access Driver (*.mdb)};DriverId=25"
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open DSN_BASE
Set Rs = Server.CreateObject("ADODB.Recordset")
```

On définit une requête SQL, et on l'exécute, que l'on nomme RequeteSQL, qui utilise la fonction "UPDATE" pour enregistrer les modifications des données dans la table (cette fonction modifie une ligne déjà existante) :

- après SET, le nom du champ dans la table suivi de sa nouvelle valeur :

```
RequeteSQL = "Update recompenses Set lot=' " & lot_bis & " ', sponsor=' " & sponsor_bis & " ',  
description=' " & description_bis & " ', place_lot=' " & placelot_bis & " ' WHERE ref_lot=' " & RefLot  
Rs.Open RequeteSQL,CONN
```

- on précise avec WHERE la ligne concernée par la modification

On vide le recordset ; on ferme et on vide la connexion.

```
set Rs=nothing  
Conn.close  
Set Conn=nothing
```

Et voilà, les modifications sont terminées!

Il ne reste plus qu'à laisser un message de confirmation pour signaler que tout s'est bien passé! C'est du code HTML, mais on peut y mêler de l'ASP :

```
<DIV align="center">
```

Merci! Le lot **<%=lot_bis%>** a bien été modifié!

```
</DIV>
```

Remarque :

Dans les fichiers téléchargeables avec cet exercice, vous verrez que nous avons utilisé, avant chaque nouvelle insertion dans la table, la commande **REPLACE**. Cette commande permet de remplacer les caractères inclus dans une variables qui ne seraient pas compatibles avec l'ASP, tels <, >, ou encore l'apostrophe et le retour chariot.

REPLACE remplace ces caractères par leur équivalent HTML.



www.asp-irine.com