

Lecture dans une base de données



Les lots, les participants, voire les arrivées viennent d'être validés... Pour le moment, le seul moyen de vérifier que l'enregistrement dans la base a bien pu être effectué revient à ouvrir cette base sous access, et à déchiffrer, ligne par ligne, chacun des enregistrements.

La manipulation est contraignante, d'autant que pendant ce temps, pendant que la base est ouverte, aucun nouvel enregistrement ne peut être effectué!

Pour éviter cela, nous allons utiliser la programmation ASP pour lister le contenu de chaque table.

[Comment lister le contenu des tables?](#) [Le mettre en forme?](#) [Filtrer les enregistrements?](#)

Préparation des fichiers

Il faut que nous puissions lister les lots, les participants, et la grille d'arrivée de la course. Au total, trois fichiers doivent être créés :

Fichiers	rôle
liste_lots.asp	lister tous les cadeaux offerts pour la course
liste_participants.asp	lister les participants
grille_arrivee.asp	visualiser l'arrivée de la course

Pour chacun de ces fichiers, nous verrons d'abord comment lister **tout le contenu** de la table correspondante ; ensuite, il faudra rajouter des **critères de sélection** (par exemple, quels sont les cadeaux offerts par Air France?).

l'exemple qui suit est réalisé à partir du fichier "liste_lots.asp" ; il s'adapte ligne par ligne aux deux autres fichiers!

Afficher tout le contenu d'une table

Pour l'écriture dans une table, nous ouvrons un recordset en exécutant la requête "Insert into".

Ici, la procédure est plus simple encore : le recordset va contenir une **sélection** que nous pouvons définir. Il suffira, ensuite, d'afficher son contenu pour afficher les enregistrements de la table sélectionnée!

 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 ConnAfficheLots = Server.CreateObject("ADODB.Connection")

ConnAfficheLots.Open DSN_BASE

Set RsAfficheLots = Server.CreateObject("ADODB.Recordset")
```

 **ouverture** du recordset :

```
RsAfficheLots.Open "Select * from recompenses" , ConnAfficheLots
```

à partir de la connexion définie, on sélectionne tous les champs (*) de la table récompenses :
en principe, le recordset parcourt la table, et se positionne sur le premier enregistrement trouvé.

 **gestion d'une erreur** fréquente : EOF=BOF

Cette erreur signifie que la table ne contient aucun enregistrement ; le recordset ne sait pas où se positionner, d'où le message qui signifie que le début et la fin du fichier sont au même endroit (table vide). Pour éviter ce problème, il faut préciser que tout ce qui va suivre cette ligne de commande ne se déroulera que si le recordset trouve au moins un enregistrement :

```
if rsAfficheLots.eof = false then
```

- ce qui signifie : "si le recordset n'arrive pas à la fin du fichier, alors..." ;
- ce qui sous-entend : "sinon va directement au END IF qui suit, sans exécuter les lignes de programmes".

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

(se positionner sur le premier enregistrement trouvé)

```
rsAfficheLots.MoveFirst
```

(début de la boucle : "tant qu'on n'est pas à la fin du fichier, alors...")

```
do while not  
rsAfficheLots.eof
```

(on affiche le contenu de l'enregistrement sur lequel est positionné le curseur - ou recordset, par champ de la table)

(ici, le nom du lot en question)


```
=rsAfficheLots("lot")
```

(se positionner sur l'enregistrement suivant)


```
rsAfficheLots.MoveNext
```

(fin de la boucle, on retourne à la ligne "do while...")

```
loop
```

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

```
end if
```

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


```
rsAfficheLots.Close
Set rsAfficheLots = Nothing
ConnAfficheLots.Close
Set ConnAfficheLots = Nothing
```

Mise en forme du résultat :

Tout ce que nous avons fait jusqu'à maintenant, c'est lister le contenu d'une table, sans mise en forme.

Résultat, tout s'affiche à la suite - ou, au mieux, en colonne si l'on a pensé à mettre la balise `
` devant chaque affichage `<%=rsAfficheLots("lot")%>`.

Ici, nous allons voir comment profiter au mieux de la boucle **Do While**.

 **Exemple** : l'objectif serait d'obtenir le **tableau** suivant.

lot	sponsor	description	place
voyage aux Baléares	Air France	une semaine pour deux (etc...)	1
week-end à Venise	Air France	pour deux (etc...)	2

C'est à dire :

- une ligne de titre
- une ligne supplémentaire par enregistrement trouvé dans la table "recompenses".

"A chaque fois que l'on trouve un enregistrement, on rajoute une ligne"

... quand on sait que les lignes se définissent en HTML par les balises `<tr>` `</tr>`, il suffit de dessiner la boucle autour :

```
<table>
<%'-----définition de la ligne des titres de colonnes-----%>
<tr>
  <td>lot</td>
  <td>sponsor</td>
  <td>description</td>
  <td>position</td>
</tr>
<%'-----affichage des résultats, une ligne par enregistrement-----%>
```

```

<%rsAfficheLots.MoveFirst
DO WHILE not rsAfficheLots.eof%>
  <tr>
    <td><B><%=rsAfficheLots("lot")%></B></td>
    <td><%=rsAfficheLots("sponsor")%></td>
    <td><%=rsAfficheLots("description")%></td>
    <td><%=rsAfficheLots("place_lot")%></td>
  </tr>
  <%rsAfficheLots.MoveNext
loop%>
<%'----- Fin de la boucle, et de la ligne----- %>
</table>

```

La même ligne, mise en forme de la même manière (ici, avec le nom du lot mis en gras par les balises et), se répètera autant de fois qu'il existe d'enregistrement dans la base.

Remarque :

L'utilisation des boucles "DO WHILE" permet de comptabiliser les enregistrements sélectionnés! En effet, il suffit d'initialiser une variable numérique (que l'on appelle par exemple compteur) :

```
compteur=0
```

et de l'incrémenter avant de recommencer chaque boucle!

```

rs.movefirst
DO WHILE (blabla)
(...)
compteur=compteur+1
rs.movenext
loop

```


il ne reste plus qu'à l'afficher ensuite :

```

au total, <%=compteur%>
enregistrements!

```

Critères de sélection :

 **TRIS** : Comment lister le contenu de la table **par ordre alphabétique** des lots?

Pour répondre à cette question, c'est à l'ouverture du Recordset qu'il faut s'intéresser. Nous avons la ligne :

```
RsAfficheLots.Open "Select * from recompenses" , ConnAfficheLots
```