

S.G.B.D. 1

TD n° 10

Introduction à ACCESS – Modules (2)

- CORRIGE -

Objectif : travailler avec un objet Recordset pour lire séquentiellement des enregistrements, utiliser la méthode Print d'un objet Report lorsqu'un événement Impression se produit.

Note : les exercices utilisent la BD MICRO sur [\\chopin\prof\dn\pub\MICRO\MDB](http://chopin.prof.dn.pub\MICRO\MDB), la copier dans votre répertoire de travail.

1. Travailler avec un objet Recordset

L'objet **Recordset** permet de lire séquentiellement les enregistrements résultant de l'exécution d'une requête Sélection. Les principales propriétés et méthodes d'un objet Recordset sont données en Annexe 1.

- 1.1. Créer une procédure LireMarques() qui remplit un tableau ListeMarques avec les différentes marques de la table Produit, et ensuite affecte la variable NbMarques avec le nombre de marques. Cette procédure ainsi que les deux variables ListeMarques et NbMarques seront déclarés dans un module Général que vous devez créer.

```
Dim ListeMarques(100) As String 'tableau des marques
Dim NbMarques As Integer       'nombre de marques

Private Sub LireMarques()
    Dim bds As Database
    Dim chSQL As String
    Dim cursV As Recordset
    Dim i As Integer

    'construire la requete SQL
    chSQL = "SELECT DISTINCT P.Marque FROM Produit P;"

    Set bds = CurrentDb()

    i = 0
    Set cursV = bds.OpenRecordset(chSQL, dbOpenSnapshot)
    Do Until cursV.EOF
        ListeMarques(i) = cursV!Marque
        Debug.Print cursV!Marque
        i = i + 1
        cursV.MoveNext           'se deplacer a l'enregistrement suivant
    Loop
    NbMarques = cursV.RecordCount
    Debug.Print NbMarques
    cursV.Close

    Set bds = Nothing
End Sub
```

- 1.2. Tester la procédure LireMarques() dans la fenêtre de débogage à l'aide de l'instruction : Call Général.LireMarques().

2. Utiliser la méthode Print d'un objet Report

La méthode **Print** d'un objet **Report** permet d'imprimer du texte lorsque l'événement **Impression** (Print) se produit. Vous utiliserez la propriété **FontBold** pour spécifier si vous désirez qu'une police apparaisse en gras

lors de l'utilisation de la méthode Print. Les principales propriétés et méthodes d'un objet Report sont données en Annexe2.

- 2.1. Nous voulons imprimer pour chaque client dans la table Client, la liste des ventes de ce client en faisant apparaître en gras les lignes dont le montant est supérieur à 5000 F, exemple ci-dessous.

1	Leblanc	Nice					
	01/04/97	10	ps	ibm	5 000,00F	1	5 000,00F
	01/04/97	20	mac	apple	7 500,00F	7	52 500,00F
	15/04/97	30	aptiva	ibm	12 000,00F	1	12 000,00F
	15/04/97	40	power mac	apple	20 000,00F	1	20 000,00F
	30/05/97	50	word	microsoft	1 000,00F	1	1 000,00F
2	Levert	Paris					
	01/04/97	10	ps	ibm	5 000,00F	1	5 000,00F
	08/05/97	20	mac	apple	7 500,00F	1	7 500,00F
	08/05/97	30	aptiva	ibm	12 000,00F	1	12 000,00F
	30/05/97	70	paradox	borland	800,00F	1	800,00F

L'état sera basé sur la table Client. Vous utiliserez une procédure événementielle lorsque l'événement Impression se produit à chaque changement de client. Cette procédure ouvrira un Recordset correspondant à l'ensemble des lignes ventes du client en cours, puis le lira séquentiellement pour imprimer les lignes ventes en spécifiant une police en gras si le montant de la vente dépasse 5000 F.

Procéder en plusieurs étapes :

1. Créer un nouvel état en mode Création
2. Définir la table Client comme Source des données
3. Dans Détail placer les zones de texte associées aux champs IdCli, Nom et Ville de la table Client
4. Créer la procédure événementielle Sur impression qui sera déclenchée à chaque client.

```
Private Sub Détail_Print(Cancel As Integer, PrintCount As Integer)
    Dim bds As Database
    Dim chSQL As String
    Dim cursV As Recordset
    Dim cm, y0, y As Integer

    'construire la requete SQL
    chSQL = "SELECT V.Date, V.IdPro, P.Désignation, P.Marque, P.PrixUht, V.Qte "
    chSQL = chSQL + "FROM Vente V INNER JOIN Produit P ON V.IdPro=P.IdPro "
    chSQL = chSQL + "WHERE V.IdCli=" + Str(IdCli) + " "
    chSQL = chSQL + "ORDER BY V.Date, V.IdPro ;"

    Set bds = CurrentDb()

    Me.FontName = "Arial"      ' police Arial
    Me.FontSize = 8           ' taille de 8 points

    cm = 567                  '1 cm = 567 twips = 28 points

    y0 = 0.5 * cm             ' position verticale premiere ligne
    y = y0                     ' position verticale ligne courante

    Set cursV = bds.OpenRecordset(chSQL, DB_OPEN_SNAPSHOT)
    Do Until cursV.EOF
        If cursV!Qte * cursV!PrixUht > 5000 Then
            Me.FontBold = True
        Else
            Me.FontBold = False
        End If

        Me.CurrentY = y

        Me.CurrentX = 1.5 * cm
        Me.Print cursV!Date;
        Me.CurrentX = 3.5 * cm
        Me.Print cursV!IdPro;
        Me.CurrentX = 4.5 * cm
        Me.Print cursV!Désignation;
        Me.CurrentX = 7 * cm
        Me.Print cursV!Marque;
        Me.CurrentX = 11.5 * cm - Me.TextWidth(Format(cursV!PrixUht, "### ##0.00\F"))
        Me.Print Format(cursV!PrixUht, "### ##0.00\F");
    Loop
End Sub
```

```

    Me.CurrentX = 12 * cm
    Me.Print cursV!Qte;
    Me.CurrentX = 14.5 * cm - Me.TextWidth(Format(cursV!PrixUht * cursV!Qte, "###
##0.00\F"))
    Me.Print Format(cursV!PrixUht * cursV!Qte, "### ##0.00F");

    y = y + 10 * 567 / 28 ' interligne de 10 points
    cursV.MoveNext      'se déplacer a l'enregistrement suivant
Loop
cursV.Close

Set bds = Nothing
End Sub

```

3. Annexe 1 – L’objet Recordset

Un objet **Recordset** représente les enregistrements d'une table de base ou les enregistrements résultant de l'exécution d'une requête. Le terme Recordset se traduit par « jeu d'enregistrement » ou « ensemble d'enregistrement ». Un Recordset permet de lire des enregistrements en accès séquentiel à l'intérieur d'une procédure Access Basic.

Il existe deux types d'objet Recordset : Feuille de réponse dynamique (**Dynaset**) et Instantané (**Snapshot**).

Un Recordset de type Feuille de réponse dynamique (Dynaset) est une sélection *dynamique* d'enregistrements. cela signifie que les enregistrements peuvent être mis à jour (modification, ajout ou suppression).

Un Recordset de type Instantané (Snapshot) est une copie statique d'un jeu d'enregistrement qui ne peut pas être mis à jour.

Créer un Recordset

Un objet Recordset est créé à l'aide de la méthode **OpenRecordset** d'un objet Database.

```

Set recordset = object.OpenRecordset (source, type)
recordset.... variable objet représentant l'objet Recordset à ouvrir.
object..... variable objet représentant un objet Database existant.
source..... instruction SQL qui renvoie des enregistrements
type ..... type d'objet Recordset à ouvrir (dbOpenDynaset ou dbOpenSnapshot)

```

Lire séquentiellement un Recordset

La méthode **MoveNext** permet de se déplacer à l'enregistrement suivant dans un Recordset (on dit aussi que l'on déplace le pointeur d'enregistrement).

Détecter la fin d'un Recordset

La propriété **Eof** prend la valeur True lorsque la fin d'un Recordset est atteinte. Lorsque vous créez un Recordset, l'enregistrement en cours est positionné sur le premier enregistrement si des enregistrements existent, sinon la propriété **RecordCount** prend la valeur 0 et Eof prend la valeur True.

4. Annexe 2 – L’objet Report

Un objet **Report** fait référence à un état Microsoft Access particulier. L'état courant peut être désigné par la variable objet **Me**. Vous disposez de méthodes et propriétés pour imprimer du texte, définir la position courante d'impression, choisir la police, la taille et le style des caractères.

Imprimer du texte

La méthode **Print** de l'objet Report permet d'imprimer une chaîne de caractères.

```

Ex. :
Me.Print "première ligne"
Me.Print "deuxième " ;
Me.Print "ligne"

```

```
Me.Print "troisième ligne"
```

Un ordre Print ne passe pas à la ligne suivante s'il a été précédé d'un autre ordre Print se terminant par un ";".

Définir la position courante d'impression

Les propriétés **CurrentX** et **CurrentY** permettent de spécifier les coordonnées du coin supérieur gauche du rectangle qui contiendra les caractères imprimés. Les coordonnées sont exprimées en *Twips*.

1 cm = 567 twips = 28 points

L'origine des coordonnées est le coin supérieur gauche d'une page.

Ex. :

```
Me.CurrentX = 3000
Me.CurrentY = 2000
Me.Print "Bonjour"
```

Connaître la largeur d'une chaîne

La méthode **TextWidth** permet de connaître la largeur d'une chaîne dans la police courante.

Ex. :

```
Me.TextWidth("Bonjour")
```

Imprimer des nombres

La fonction **Format** permet de spécifier un masque d'impression :

0	Réserve l'emplacement d'un chiffre
#	Réserve l'emplacement d'un chiffre s'il ne s'agit pas d'un zéro à gauche, sinon un blanc
.	Réserve le caractère séparant la partie entière et la partie décimale
,	Réserve le caractère séparant les milliers
\	Indique que les caractères qui suivent sont imprimés tels quels

Ex. :

```
Prix = 2312.5
PrixF = Format(Prix, "##,##0.00\F")
```

Pour cadrer un nombre à droite, il faut calculer la position de gauche = Position de droite - Largeur du nombre

Choisir la police, la taille, le style des caractères

Pour les choisir, vous disposez des propriétés suivantes :

FontName	Nom de la police, "Arial ", "Times New Roman" ...
FontSize	Taille des caractères en points
FontBold	True pour des caractères gras ou False sinon
FontItalic	True pour des caractères italique ou False sinon
FontUnderline	True pour des caractères souligné ou False sinon

Ex. :

```
Me.FontName = "Arial"
Me.FontSize = 14
Me.FontBold = True
Me.FontItalic = True
Me.Print "Voici un texte en Arial de 14, en gras et italique"
```