

Compte rendu semaine 1

Formulaire d'ajout d'un club

Aspect graphique

Dressage des contours

Nous commençons tout d'abord par créer un nouveau formulaire vierge, puis nous établissons la structure du formulaire, tout en respectant la charte graphique prédéfinie au préalable par toute l'équipe, ce qui implique le respect de la taille du formulaire et des différents cadres, ainsi que des différentes couleurs utilisées.

Mise en place des éléments de texte

Nous remplissons ensuite le cadre du haut avec une étiquette contenant le titre du formulaire (ici « GESTIONS DS CLUBS [AJOUTER] ») puis le cadre du milieu avec des zones de texte, afin de saisir les informations concernant les différents champs de la table CLUB.

Ajout des boutons

Ils nous reste ensuite à inclure le bouton « AJOUTER » qui nous permettra de créer un nouvel enregistrement dans la table CLUB ainsi que le bouton « FERMER » qui nous permettra tout simplement de fermer le formulaire.

Aspect programmation

Une fois l'aspect graphique terminé nous passons donc au code sur chaque bouton. Pour commencer par le plus simple, le bouton FERMER, sert à fermer le formulaire. Ainsi la simple ligne de code suivante suffit :

```
Private Sub Cmd_Fermer_Click()  
DoCmd.Close  
End Sub
```

En revanche, le code du bouton AJOUTER va être quelque peu plus complexe. En effet, celui ci va comprendre plusieurs tests. Tout d'abord, nous voulons que chaque champ du formulaire soit obligatoirement rempli. Pour cela, il nous suffit de vérifier si le champ contient une valeur nulle ou non. Dans le cas où le champ est vide, nous retournons un message d'erreur via MsgBox, sinon on

passer au test suivant. Voici le code concernant la vérification de la zone de texte pour le numéro du club :

```
If IsNull(Txt_Num) Then
    MsgBox "Saisir un numéro", vbExclamation, "Gestion Club"
    Txt_Num.SetFocus 'positionner le curseur sur le champ txt_Num
    Exit Sub 'Quitter la procédure
End If
```

Ensuite nous vérifions si le numéro de club, qui est clé primaire dans la table club, n'existe pas déjà. Etant donné que c'est une clé primaire celui ci doit être unique.

```
Dim dbClub As Database
Dim cClub As DAO.Recordset
Dim sql As String
sql = "select NUM_CLUB from CLUB where NUM_CLUB=" & Txt_Num
Set dbClub = CurrentDb()
Set cClub = dbClub.OpenRecordset(sql, DB_OPEN_DYNASET)
If cClub.EOF() = False Then
    MsgBox "Le numéro existe déjà", vbCritical, "Gestion Club"
    Txt_Nom.SetFocus
    Exit Sub
End If
```

Enfin, si le Club n'existe pas déjà, nous passons donc à la création de celui ci à l'aide d'un .AddNew puis d'un .Update afin de valider/mettre à jour les nouvelles informations.

```
Set cClub = CurrentDb().OpenRecordset("CLUB")
cClub.AddNew
cClub!NUM_CLUB = Txt_Num
cClub!NOM_CLUB = Txt_Nom
cClub!ADR_RUE_CLUB = Txt_Adresse
cClub!CODE_POST_CLUB = Txt_CodePostal
cClub!ADR_VILLE_CLUB = Txt_Ville
cClub.Update
```

Ensuite, nous affichons un message de confirmation via MsgBox et pour une éventuelle nouvelle création à la suite de celle ci, nous remettons les champs à vide.

```
MsgBox("La création a bien été effectuée")

Txt_Num = "": Txt_Nom = "":
Txt_Adresse = "": Txt_CodePostal = "":
Txt_Ville = ""
```

Le code final est donc celui ci :

```
Private Sub Cmd_Ajouter_Click()
If IsNull(Txt_Num) Then
    MsgBox "Saisir un numéro", vbExclamation, "Gestion Club"
```

```
Txt_Num.SetFocus 'positionner le curseur sur le champ txt_Num
Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_Nom) Then
    MsgBox "Saisir un nom", vbExclamation, "Gestion Club"
    Txt_Num.SetFocus 'positionner le curseur sur le champ txt_Num
    Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_Adresse) Then
    MsgBox "Saisir une adresse", vbExclamation, "Gestion Club"
    Txt_Num.SetFocus 'positionner le curseur sur le champ txt_Adresse
    Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_CodePostal) Then
    MsgBox "Saisir un code postal", vbExclamation, "Gestion Club"
    Txt_Num.SetFocus 'positionner le curseur sur le champ txt_CodePostal
    Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_Ville) Then
    MsgBox "Saisir un nom de ville", vbExclamation, "Gestion Club"
    Txt_Num.SetFocus 'positionner le curseur sur le champ txt_Ville
    Exit Sub 'Quitter la procédure
End If

Dim dbClub As Database
Dim cClub As DAO.Recordset
Dim sql As String
sql = "select NUM_CLUB from CLUB where NUM_CLUB=" & Txt_Num
Set dbClub = CurrentDb()
Set cClub = dbClub.OpenRecordset(sql, DB_OPEN_DYNASET)
If cClub.EOF() = False Then
    MsgBox "Le numéro existe déjà", vbCritical, "Gestion Club"
    Txt_Num.SetFocus
    Exit Sub
End If

Set cClub = CurrentDb().OpenRecordset("CLUB")
cClub.AddNew
cClub!NUM_CLUB = Txt_Num
cClub!NOM_CLUB = Txt_Nom
cClub!ADR_RUE_CLUB = Txt_Adresse
cClub!CODE_POST_CLUB = Txt_CodePostal
cClub!ADR_VILLE_CLUB = Txt_Ville
cClub.Update

MsgBox("La création a bien été effectuée")
```

```
Txt_Num = "": Txt_Nom = "":  
Txt_Adresse = "": Txt_CodePostal = "":  
Txt_Ville = ""  
End Sub
```

Compte rendu semaine 2

Formulaire de modification/suppression d'un club

Aspect graphique

Nous allons ensuite créer un formulaire vierge pour la modification et suppression. Comme pour la semaine dernière nous allons établir la structure du formulaire, tout en respectant la charte graphique prédéfinie au préalable par toute l'équipe, ce qui implique le respect de la taille du formulaire et des différents cadres, ainsi que des différentes couleurs utilisées.

Comme vous le voyez ci-dessus, nous avons choisi une zone de liste pour le choix du club afin d'afficher tous les données dans les champs concernés, ceci à un aspect plus pratique. Et trois boutons pour le traitement voulu.

Aspect programmation

Tout d'abord sur la liste, nous exécutons un code VBA, qui nous permet tout simplement grâce à une requête SQL de récupérer les informations manquantes du club choisi dans la liste déroulante

```
Private Sub lst_Club_Change()  
  
Dim dbClub As Database  
Dim rsClub As Recordset  
Dim requete As String  
  
Set dbClub = CurrentDb()  
requete = "SELECT * FROM CLUB WHERE NUM_CLUB=" & lst_Club.Column(0)  
Set rsClub = dbClub.OpenRecordset(requete, DB_OPEN_DYNASET)  
  
'Afficher les données concernant le club  
  
Txt_Num = rsClub!NUM_CLUB  
Txt_Nom = rsClub!NOM_CLUB  
Txt_Adresse = rsClub!ADR_RUE_CLUB  
Txt_CodePostal = rsClub!CODE_POST_CLUB  
Txt_Ville = rsClub!ADR_VILLE_CLUB
```

End Sub

Modification d'un enregistrement

Ensuite, concernant le bouton Modifier, nous vérifions comme pour le bouton AJOUTER du premier formulaire, que les champs sont bien tous remplis et qu'il n'en reste pas un de vide.

```
Private Sub Cmd_Modifier_Click()

'----- gestion des erreurs de saisie

If IsNull(Txt_Nom) Then
    MsgBox "Saisir un nom", vbExclamation, "Gestion CLUB"
    Txt_Nom.SetFocus 'positionner le curseur sur le champ txt_Nom
    Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_Adresse) Then
    MsgBox "Saisir une adresse", vbExclamation, "Gestion CLUB"
    Txt_Adresse.SetFocus 'positionner le curseur sur le champ
    Exit Sub 'Quitter la procédure
End If

If IsNull(Txt_CodePostal) Then
    MsgBox "Saisir un code postal", vbExclamation, "Gestion CLUB"
    Txt_CodePostal.SetFocus
    Exit Sub
End If

If IsNull(Txt_Ville) Then
    MsgBox "Saisir une ville", vbExclamation, "Gestion CLUB"
    Txt_Ville.SetFocus
    Exit Sub
End If
```

Une fois cette batterie de test effectuée, nous passons au contrôle de la base de données.

```
'----- contrôle base de données
Dim dbClub As Database
Dim rsClub As Recordset
Dim requete As String

Set dbClub = CurrentDb()
requete = "SELECT * FROM CLUB WHERE NUM_CLUB=" & lst_Club.Column(0)
Set rsClub = dbClub.OpenRecordset(requete, DB_OPEN_DYNASET)
```

Il ne nous reste plus qu'à modifier les champs qui l'ont été, à l'aide d'un rs.Edit, qui permet de modifier les enregistrement dans la table voulue, ici CLUB.

```
'----- modifier un club dans la table CLUB
```

```
rsClub.Edit
```

```
rsClub!NOM_CLUB = Txt_Nom
```

```
rsClub!ADR_RUE_CLUB = Txt_Adresse
```

```
rsClub!CODE_POST_CLUB = Txt_CodePostal
```

```
rsClub!ADR_VILLE_CLUB = Txt_Ville
```

```
rsClub.Update
```

Ensuite pour la mise en forme, nous affichons un message de confirmation à l'aide d'un MsgBox, puis nous remettons les champs à vide afin de poursuivre vers une éventuelle seconde modification ou suppression.

```
MsgBox ("La modification est un succès")
```

```
'initialiser les champs
```

```
lst_Club = ""
```

```
Txt_Num = ""
```

```
Txt_Nom = ""
```

```
Txt_Adresse = ""
```

```
Txt_CodePostal = ""
```

```
Txt_Ville = ""
```

```
End Sub
```

Le code final est donc celui-ci :

```
Private Sub Cmd_Modifier_Click()
```

```
'----- gestion des erreurs de saisie
```

```
If IsNull(Txt_Nom) Then
```

```
    MsgBox "Saisir un nom", vbExclamation, "Gestion CLUB"
```

```
    Txt_Nom.SetFocus 'positionner le curseur sur le champ txt_Nom
```

```
    Exit Sub 'Quitter la procédure
```

```
End If
```

```
If IsNull(Txt_Adresse) Then
```

```
    MsgBox "Saisir une adresse", vbExclamation, "Gestion CLUB"
```

```
    Txt_Adresse.SetFocus 'positionner le curseur sur le champ
```

```
    Exit Sub 'Quitter la procédure
```

```
End If
```

```
If IsNull(Txt_CodePostal) Then
```

```
    MsgBox "Saisir un code postal", vbExclamation, "Gestion CLUB"
```

```
    Txt_CodePostal.SetFocus
```

```
    Exit Sub
```

```
End If
```

```
If IsNull(Txt_Ville) Then
```

```
    MsgBox "Saisir une ville", vbExclamation, "Gestion CLUB"
    Txt_Ville.SetFocus
    Exit Sub
End If

'----- contrôle base de données
Dim dbClub As Database
Dim rsClub As Recordset
Dim requete As String

Set dbClub = CurrentDb()
requete = "SELECT * FROM CLUB WHERE NUM_CLUB=" & lst_Club.Column(0)
Set rsClub = dbClub.OpenRecordset(requete, DB_OPEN_DYNASET)

'----- modifier un club dans la table CLUB

rsClub.Edit

rsClub!NOM_CLUB = Txt_Nom
rsClub!ADR_RUE_CLUB = Txt_Adresse
rsClub!CODE_POST_CLUB = Txt_CodePostal
rsClub!ADR_VILLE_CLUB = Txt_Ville
rsClub.Update

MsgBox ("La modification est un succès")

'initialiser les champs
lst_Club = ""
Txt_Num = ""
Txt_Nom = ""
Txt_Adresse = ""
Txt_CodePostal = ""
Txt_Ville = ""

End Sub
```

Suppression d'un enregistrement

Nous allons retrouver beaucoup de points communs avec la modification au niveau du code, hormis le fait qu'ici, la requête a pour but de SUPPRIMER un enregistrement, d'où la requête "DELETE...FROM"

```
Private Sub Cmd_Supprimer_Click()
Dim bdClub As Database
Dim rsClub As Recordset
Dim cCompet As Recordset
Dim cMembre As Recordset
Dim requete As String
Dim requete1 As String
Dim requete2 As String
```

```
Set bdClub = CurrentDb()

requete1 = "SELECT NUM_CLUB FROM COMPETITION WHERE NUM_CLUB=" & Txt_Num
Set cCompet = bdClub.OpenRecordset(requete1, DB_OPEN_DYNASET)

If Not cCompet.EOF() Then
    MsgBox "Vous ne pouvez pas supprimer le Club car des compétitions sont
référencées pour ce club", vbExclamation, "Gestion CLUB"
    lst_Club.SetFocus
Exit Sub
End If

requete2 = "SELECT NUM_CLUB FROM MEMBRE WHERE NUM_CLUB=" & Txt_Num
Set cMembre = bdClub.OpenRecordset(requete2, DB_OPEN_DYNASET)

If Not cMembre.EOF() Then
    MsgBox "Vous ne pouvez pas supprimer le Club car des membres sont
référencés pour ce club", vbExclamation, "Gestion CLUB"
    lst_Club.SetFocus
Exit Sub
End If
```

Ensuite pour supprimer le club grâce à la requête présente ci dessus, nous utilisons un DoCmd.RunSQL afin d'exécuter la requête préalablement établie, à une condition : Que l'enregistrement que l'on veut supprimer ne fasse pas référence à un autre champ dans une autre table !

```
'suppression du club

requete = "DELETE * FROM club WHERE NUM_CLUB=" & Txt_Num
DoCmd.RunSQL (requete)
```

Ici comme pour au dessus, nous affichons un message de confirmation et remettons les champs à vide.

```
MsgBox ("Le club a été supprimé avec succès")

'initialiser les champs

lst_Club = ""
Txt_Num = ""
Txt_Nom = ""
Txt_Adresse = ""
Txt_CodePostal = ""
Txt_Ville = ""

End Sub
```

Le code final est donc celui-ci :

Option Explicit

```
Private Sub Cmd_Supprimer_Click()  
Dim bdClub As Database  
Dim rsClub As Recordset  
Dim cCompet As Recordset  
Dim cMembre As Recordset  
Dim requete As String  
Dim requete1 As String  
Dim requete2 As String  
  
Set bdClub = CurrentDb()  
  
requete1 = "SELECT NUM_CLUB FROM COMPETITION WHERE NUM_CLUB=" & Txt_Num  
Set cCompet = bdClub.OpenRecordset(requete1, DB_OPEN_DYNASET)  
If Not cCompet.EOF() Then  
    MsgBox "Vous ne pouvez pas supprimer le Club car des compétitions sont  
référencées pour ce club", vbExclamation, "Gestion CLUB"  
    lst_Club.SetFocus  
Exit Sub  
End If  
  
requete2 = "SELECT NUM_CLUB FROM MEMBRE WHERE NUM_CLUB=" & Txt_Num  
Set cMembre = bdClub.OpenRecordset(requete2, DB_OPEN_DYNASET)  
If Not cCompet.EOF() Then  
    MsgBox "Vous ne pouvez pas supprimer le Club car des membres sont  
référencées pour ce club", vbExclamation, "Gestion CLUB"  
    lst_Club.SetFocus  
Exit Sub  
End If  
  
'suppression club  
  
requete = "DELETE * FROM club WHERE NUM_CLUB=" & Txt_Num  
DoCmd.RunSQL (requete)  
  
MsgBox ("Le club a été supprimé avec succès")  
  
lst_Club = ""  
Txt_Num = ""  
Txt_Nom = ""  
Txt_Adresse = ""  
Txt_CodePostal = ""  
Txt_Ville = ""  
  
End Sub
```

Liste des membres par club

Aspect Graphique

Pour ce formulaire nous avons décidé, comme pour le précédent formulaire, de mettre une liste déroulante pour le choix des clubs. Concernant l'affichage des membres selon le club choisi, nous avons utilisé une zone de liste affichant le numéro de licence, le nom, le prénom, la date de naissance ainsi que l'adresse de chaque membre du club concerné. Concernant la mise en forme de ce formulaire nous avons repris la charte graphique définie, et utilisée dans les précédents formulaires.

Aspect programmation

Afin d'afficher la liste des membres appartenant au club choisi, le code est similaire à celui de la liste déroulante du formulaire de modification/suppression à quelques différences près. Tout d'abord dans la requête, nous devons triés les membres choisis par année de naissance et pas nom. Nous avons donc ajouté la clause "ORDER BY" avec la fonction YEAR, qui extrait uniquement l'année dans les différents dates de naissance.

```
Private Sub lst_Club_Change()  
    Dim dbClub As Database  
    Dim rsClub As Recordset  
    Dim requete As String  
    Set dbClub = CurrentDb()  
    requete = "SELECT NUM_LICENCE, NOM_MEMBRE, PRENOM_MEMBRE,  
DATE_NAISSANCE, ADR_RUE_MEMBRE FROM MEMBRE WHERE NUM_CLUB = " & lst_Club & "  
ORDER BY  
    year(DATE_NAISSANCE), NOM_MEMBRE "
```

Ensuite, afin de remplir la zone de liste directement lors du choix du club, nous utilisons la fonction RowSource, qui nous permet de remplir des listes à l'aide d'une requête. Ici la requête est celle définie ci dessus.

```
'Actualiser la liste des membres  
lst_Membre.RowSource = requete  
lst_Membre.Requery
```

```
End Sub
```

Pour le code du bouton fermer, c'est exactement le même que celui du formulaire pour ajouter un club, un simple DoCmd.Close.

Compte rendu semaine 3 & 4

Formulaire MENU

Aspect Graphique

Concernant l'aspect graphique du menu principal, nous l'avons personnalisé afin qu'il soit plus agréable à l'œil que les formulaires de saisie ou de modification d'informations. Voici donc à quoi ressemble le menu principal.

Concernant la présentation des sous-menus correspondant aux tâches à effectuer selon le choix fait au menu principal, nous avons repris la même charte graphique que le menu principal avec une disposition différente des boutons. Voici donc à quoi ressemble le sous-menu "JUGE" avec les différentes fonctions que nous pouvons effectuer sur la table "JUGE".

Aspect programmation

Le menu principal et les sous-menus ne comportent pas de code VBA, mais simplement des macros permettant de naviguer entre les différents menus et formulaires. La création de macro est rendue beaucoup plus facile via l'assistant intégré à Access. Voici à quoi il ressemble :

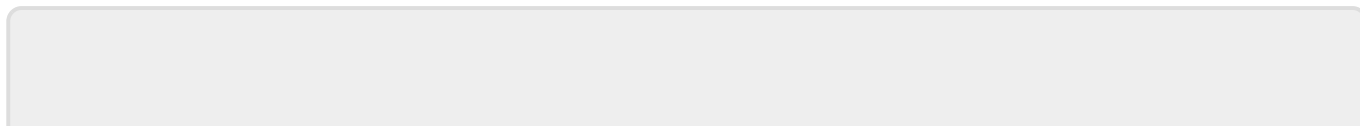
Nous pouvons apercevoir 2 colonnes : Action et Arguments.

La colonne Action sert à déterminer ce à quoi va servir la macro. En l'occurrence, ici, la macro va nous servir à ouvrir le formulaire approprié selon la fonction que l'on aura choisie.

Exemple : Si l'on sélectionne le formulaire de création d'un membre, la macro va donc ouvrir le formulaire de création d'un membre.

La colonne Arguments sert à préciser certaines informations sur l'action. Par exemple, lorsqu'il s'agit d'ouvrir un formulaire, les arguments vont être le nom du formulaire, d'éventuelles conditions etc... Nous pouvons choisir ses arguments dans les champs qui s'affichent en bas de page.

Nous intégrons donc à chaque bouton d'un menu la macro permettant d'ouvrir le formulaire approprié. Sur chaque menu se trouve un bouton "FERMER" qui ferme donc le menu actuel et nous ramène au menu précédent.



From:

/ - APs et stages du BTS SIO du lycée Suzanne Valadon

Permanent link:

</doku.php/slam/ws/2011/ppe2.1/equipe2/accueil>

Last update: **2014/01/07 13:56**

