

1. AP2.1 : Projet KARATE : Mission 3

Présentation de l'équipe et de la mission	
Présentation	Nous sommes deux élèves de BTS SIO SLAM, Christian et Mélanie. Nous avons décidé de prendre la mission 3 qui concerne principalement les entraîneurs.
Tâches	Réalisation des tâches suivantes : T1.3, T2.3, T2.6.
Outils	Afin de réaliser ces tâches, nous avons créé un nouveau projet Visual Studio nommé VB.NET Karaté.

T1.3 Gestion des entraîneurs

Rappel des objectifs

Cette tâche doit permettre la création, la modification et la suppression d'un entraîneur à partir d'un formulaire. Il faudra réaliser la maquette de ce formulaire, le faire valider par le chef de projet avant de commencer le codage. Il faudra vérifier qu'un entraîneur ne peut être membre du club et qu'il n'entraîne qu'un seul club.

Formulaire de gestion des entraîneurs Création, Modification, Suppression

Comme indiqué dans la consigne nous avons préparé nos maquettes en avance afin de les faire valider par les chefs de projet.

1er maquette : Formulaire Ajouter

GESTION DES ENTRAINEURS (AJOUTER)

NUM ENTRAINEUR :

NOM :

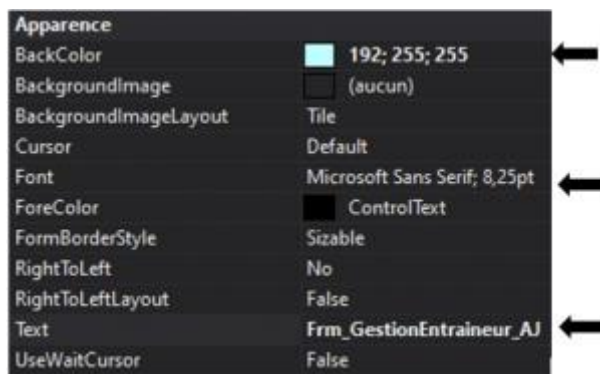
PRENOM :

DateGridView

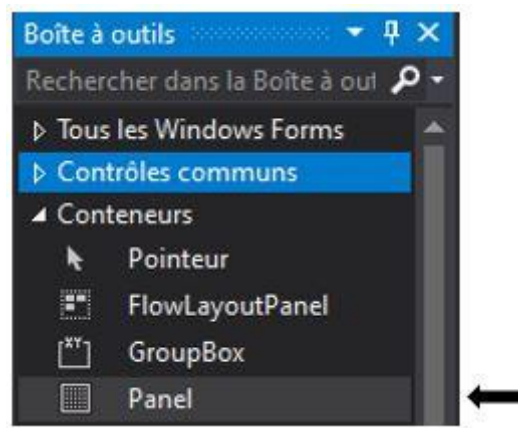
AJOUTER

FERMER

Après que la première maquette fut validée, nous avons commencé le design et le codage sur **Visual Studio**. Concernant le design nous avons fait plusieurs modifications. Nous avons ajouté le magnifique bleu dans **BackColor**, modifié la police et sa taille dans **Font** ainsi que le nom du formulaire dans **Text**.



Afin d'être conforme à la maquette proposée par les chefs de projet, nous avons utilisé la boîte à outils dans la section **Conteneurs** l'option **Panel** afin de rajouter les cadres gris.



Ensuite nous avons rajouté trois **Label**, trois **TextBox**, deux **Button** et une **DataGridView**. Ces derniers ont été renommé comme demandé dans le **Dokuwiki**.

Txt_NumEntraîneur	Txt_Prenom
Txt_Nom	Cmd_Fermer
Cmd_Ajouter	Grid_Entraîneur
Frm_GestionEntraîneur_AJ	

Aperçu du premier formulaire avant codage



Frm_GestionEntraîneur_AJ

GESTION DES ENTRAINEURS (AJOUTER)

NUM ENTRAINEUR :

NOM :

PRENOM :

	Num Club	Nom Club
*		

Tout d'abord nous avons commencé par remplir la **DataGridView**. Pour cela, nous nous sommes aidés du cours disponible sur le **Dokuwiki**.

```
Private Sub Frm_GestionEntraîneur_AJ_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
```

```
connexion("bdKarate.accdb")
```

```
Grid_Entraineur.ColumnCount = 2
```

```
Grid_Entraineur.Columns(0).Width = 90
```

```
Grid_Entraîneur.Columns(1).Width = 200
```

```
Grid_Entraîneur.Columns(0).Name = "Numéro du Club"
```

```
Grid_Entraîneur.Columns(1).Name = "Nom du Club"
```

```
Grid_Entraîneur.AllowUserToAddRows = False
```

Nous sommes sur le **Load** et nous avons fait la connexion à la base de données grâce au **MyBase.Load connexion("bdKarate.accdb")**.

La première ligne nous indique le nombre de colonne. Nous en avons deux. La deuxième et la troisième concernent la taille de la **DataGridView**. Ensuite, la quatrième et la cinquième ligne nous indique le nom de nos colonnes (les colonnes démarre à zéro).

La dernière ligne (**AllowUserToAddRows**) nous demande si nous autorisons l'utilisateur à ajouter des lignes. Dans le cas de notre formulaire nous ne l'autorisons pas, donc nous mettons **False**.

```
Dim req As String

    req = "select * from CLUB"

    Dim cmdRech As New OleDbCommand(req, maConnexion)

    Dim rdrRech As OleDbDataReader = cmdRech.ExecuteReader()
```

La première ligne concerne la requête. La deuxième permet d'envoyer la requête vers la connexion et dans la troisième ligne nous avons créé le **DataReader (rdrRech)** afin de l'exécuter.

```
Dim numLigne As Integer

    While (rdrRech.Read())

        Grid_Entraîneur.Rows.Add()

        numLigne = Grid_Entraîneur.RowCount - 1

        Grid_Entraîneur.Item(0, numLigne).Value = rdrRech.Item("NUM_CLUB")

        Grid_Entraîneur.Item(1, numLigne).Value = rdrRech.Item("NOM_CLUB")

    End While
```

Puisque notre requête retourne plusieurs enregistrements, il nous faut un curseur pour parcourir et lire tous les enregistrements. Nous avons donc utilisé un **While**.

Nous utilisons **Rows.Add**, afin d'ajouter une ligne. Puis, pour pouvoir gérer la ligne en dessous, il va se positionner sur la variable **numLigne** que nous avons précédemment déclarer avec un « **Dim maVariable As Integer** » car il s'agit d'un entier. Enfin, **RowCount** contient le nombre de lignes de la grille. Mais comme le rappelle le cour, les lignes et les colonnes commencent à 0, donc nous mettons **RowCount - 1**.

Dans les deux dernières lignes, nous utilisons l'élément **Item** car il correspond à l'intersection entre la ligne et la colonne. Puis dans le **Value** nous mettons le résultat de la requête. Pour

récupérer le résultat de la requête, il nous faut **rdrRech.Item** ainsi que les noms des champs de la table **Club** qui nous intéressent.

Et **End While** pour fermer le **While**.

Nous pouvons enfin tester notre formulaire afin de savoir si la **DataGridView** fonctionne.



	Num Club	Nom Club
▶	1	Karaté-Club Nancéien
	2	SCK Forbach
	3	Karaté-Club de Metz
	4	Shotokan-Metz

Nous allons nous occuper maintenant de la création d'un entraîneur à l'aide du bouton **AJOUTER**.

```
Private Sub Cmd_Ajouter_Click(sender As Object, e As EventArgs) Handles Cmd_Ajouter.Click

    Dim req As String

    req = "insert into ENTRAINEUR values (@num, @club, @nom, @prenom)"

    Dim cmdInsert As New OleDbCommand(req, maConnexion)
```

La deuxième ligne concerne la requête. Dans la troisième, le **insert into** permet d'insérer une ligne en indiquant les informations pour chaque colonne existante. Et la quatrième ligne permet d'envoyer la requête vers la connexion.

```
cmdInsert.Parameters.AddWithValue("@num", Txt_NumEntraîneur.Text)

cmdInsert.Parameters.AddWithValue("@club", Me.Grid_Entraîneur.Item(0,
Grid_Entraîneur.CurrentRow.Index).Value)

cmdInsert.Parameters.AddWithValue("@nom", Txt_Nom.Text)

cmdInsert.Parameters.AddWithValue("@prenom", Txt_Prenom.Text)
```

Dans les quatre lignes, nous utilisons les éléments **cmdInsert. Parameters. AddWithValue** car ils nous permettent de récupérer les résultats de la requête.

```
cmdInsert.ExecuteNonQuery()

MsgBox("Entraîneur créé...")
```

Avec la première ligne, on exécute la requête dans la base de données. Et dans la deuxième le **MsgBox** nous permettra d'afficher cette phrase lorsque nous aurons inséré l'entraîneur.

Puis nous pouvons tester.

The screenshot shows a Windows application window titled "Frm_GestionEntraîneur_AJ". The main form has a light blue background and a title bar. The title of the form is "GESTION DES ENTRAINEURS (AJOUTER)".

The form contains three text input fields with labels:

- NUM ENTRAINEUR : 7
- NOM : SUN
- PRENOM : Christian

Below these fields is a table with two columns: "Num Club" and "Nom Club". The table has four rows:

Num Club	Nom Club
1	Karaté-Club Nancéien
2	SCK Forbach
3	Karaté-Club de Metz
4	Shotokan-Metz

The third row, "Karaté-Club de Metz", is selected. Below the table are two buttons: "AJOUTER" and "FERMER".

A small dialog box titled "VB.NET Karaté" is open, showing the message "Entraîneur créé..." and an "OK" button.

Enfin, nous pouvons vérifier dans la base de données si nous retrouvons l'entraîneur ajouté.

ENTRAINEUR				
	NUM_ENTR	NUM_CLUB	NOM_ENTR	PRENOM_ENTR
+	1	1	TOURNIE	Alain
+	2	1	BOURGEOIS	Agnès
+	3	3	BOGUSZ	Thierry
+	4	3	TECHER	Charles
+	5	3	PASQUALINI	Claude
+	6	2	Dupont	Jack
+	7	3	SUN	Christian
+	123	3	TOTO	TOTO

C'est le cas 😊.

Nous n'avons plus qu'à rajouter la procédure pour fermer le formulaire.

```
Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles  
Cmd_Fermer.Click
```

```
Me.Close()
```

```
End Sub
```

2ème maquette : Formulaire Modifier/Supprimer

GESTION DES ENTRAINEURS (MODIFIER/SUPPRIMER)

NOM :

PRENOM :

DateGridView

MODIFIER

SUPPRIMER

FERMER

Après que la deuxième maquette fut validée, nous avons commencé le design et le codage sur **Visual Studio**. Ce dernier est identique à celui de la première maquette. Ensuite nous avons rajouté les composants graphiques utiles. Ces derniers ont été renommé comme demandé dans le **Dokuwiki**.

Txt_Prenom	Txt_Nom
Cmd_Modifier	Cmd_Supprimer
Cmd_Fermer	Grid_Entraîneur
Frm_GestionEntraîneur_MS	

Aperçu du deuxième formulaire avant codage

Comme pour le premier formulaire, nous avons commencé par remplir la **DataGridView**.

```
Private Sub Frm_GestionEntraîneur_MS_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    connexion("bdKarate.accdb")

    Grid_Entraîneur.ColumnCount = 3

    Grid_Entraîneur.Columns(0).Width = 200
```



```

Grid_Entraîneur.Columns(1).Width = 200

Grid_Entraîneur.Columns(2).Width = 200

Grid_Entraîneur.Columns(0).Name = "Numéro de l'entraîneur"

Grid_Entraîneur.Columns(1).Name = "Nom de l'entraîneur"

Grid_Entraîneur.Columns(2).Name = "Prénom de l'entraîneur"

Grid_Entraîneur.AllowUserToAddRows = False

Grid_Entraîneur.Columns("Numero").Visible = False

```

Dans la dernière ligne de la capture, nous avons utilisé le code « **Datagridview1. Columns ("Nom_Champs").visible = false** » afin de masquer la colonne **Numéro** car elle n'a pas besoin d'être affiché.

```

Dim req As String

req = "select * from ENTRAINEUR"

Dim cmdRech As New OleDbCommand(req, maConnexion)

Dim rdrRech As OleDbDataReader = cmdRech.ExecuteReader()

Dim numLigne As Integer

While (rdrRech.Read())

    Grid_Entraîneur.Rows.Add()

    numLigne = Grid_Entraîneur.RowCount - 1

    Grid_Entraîneur.Item(0, numLigne).Value =
rdrRech.Item("NUM_ENTRAINEUR")

    Grid_Entraîneur.Item(1, numLigne).Value =
rdrRech.Item("NOM_ENTRAINEUR")

    Grid_Entraîneur.Item(2, numLigne).Value =
rdrRech.Item("PRENOM_ENTRAINEUR")

End While

End Sub

```

Nous avons annoncé la requête puis créé le **DataReader (rdrRech)** afin de l'exécuter. Cette requête retourne elle aussi plusieurs enregistrements, il nous faut donc un curseur pour parcourir et lire tous les enregistrements.

```
Private Sub Grid_Entraîneur_CellContentClick(sender As Object, e As  
DataGridViewCellEventArgs) Handles Grid_Entraîneur.CellContentClick
```

```
    Dim numLigne As Integer
```

```
    numLigne = Grid_Entraîneur.CurrentRow.Index
```

```
    Txt_Nom.Text = Grid_Entraîneur.Item(0, numLigne).Value
```

```
    Txt_Prenom.Text = Grid_Entraîneur.Item(1, numLigne).Value
```

```
End Sub
```

Cette procédure va nous permettre d'afficher les données demandées (nom et prénom) dans nos **TextBox** lorsque nous cliquerons sur la **Grid_Entraîneur**.

Nous pouvons enfin tester notre formulaire afin de savoir si la **DataGridView** et l'affichage des données fonctionnent.

	Nom de l'entraîneur	Prenom de l'entraîneur
▶	TOURNIE	Alain
	BOURGEOIS	Agnès
	BOGUSZ	Thierry
	TECHER	Charles
	PASQUALINI	Claude
	Dupont	Jack
	TOTO	TOTO

Nous allons nous occuper maintenant de la modification d'un entraineur à l'aide du bouton **MODIFIER**.

```
Private Sub Cmd_Modifier_Click(sender As Object, e As EventArgs) Handles Cmd_Modifier.Click

    Dim req As String

    req = "UPDATE ENTRAINEUR SET NOM_ENTRAINEUR = @nom, PRENOM_ENTRAINEUR = @prenom WHERE NUM_ENTRAINEUR = @num"

    Dim cmdUpdate As New OleDbCommand(req, maConnexion)

    cmdUpdate.Parameters.AddWithValue("@nom", Txt_Nom.Text)

    cmdUpdate.Parameters.AddWithValue("@prenom", Txt_Prenom.Text)

    cmdUpdate.Parameters.AddWithValue("@num", Me.Grid_Entraîneur.Item(2, Grid_Entraîneur.CurrentRow.Index).Value)

    cmdUpdate.ExecuteNonQuery()

    MsgBox("Entraîneur modifié...")

    Me.Close()

End Sub
```

Pour modifier nous utilisons une requête **UPDTATE**. Elle va nous permettre d'attribuer une nouvelle valeur à une colonne pour les lignes qui respectent la condition stipulé avec le **WHERE**.

Ensuite, nous utilisons les éléments **cmdUpdate. Parameters. AddWithValue** car ils nous permettent de récupérer les résultats de la requête.

Il nous reste plus qu'à exécuter la requête dans la base de données. Nous choisissons d'utiliser le **MsgBox** pour afficher ce message lorsque nous aurons modifier l'entraineur.

Puis nous pouvons tester.

Form_GestionEntraîneur_MS

GESTION DES ENTRAINEURS (MODIFIER / SUPPRIMER)

NOM : PASQUALINI

PRENOM : ToutRouge

Nom de l'entraîneur	Prenom de l'entraîneur
TOURNIE	Alain
BOURGEOIS	Agnès
BOGUSZ	Thierry
TECHER	Charles
PASQUALINI	Claude
Dupont	Jack
TOTO	TOTO
SUN	Christian

VB.NET Karaté

Entraîneur modifié...

OK

MODIFIER SUPPRIMER FERMER

Enfin, nous pouvons vérifier dans la base de données si l'entraîneur a bien été modifié.

ENTRAINEUR				
	NUM_ENTR.	NUM_CLUB	NOM_ENTR.	PRENOM_EI
+	1	1	TOURNIE	Alain
+	2	1	BOURGEOIS	Agnès
+	3	3	BOGUSZ	Thierry
+	4	3	TECHER	Charles
+	5	3	PASQUALINI	ToutRouge
+	6	2	Dupont	Jack
+	7	3	SUN	Christian
+	123	3	TOTO	TOTO

C'est le cas 😊.

Enfin, nous allons nous occuper de la suppression d'un entraîneur à l'aide du bouton **SUPPRIMER**.

```

Private Sub Cmd_Supprimer_Click(sender As Object, e As EventArgs) Handles Cmd_Supprimer.Click

    Dim req As String

    req = "DELETE FROM ENTRAINEUR WHERE NOM_ENTRAINEUR = @nom and
PRENOM_ENTRAINEUR = @prenom"

    Dim cmdDelete As New OleDbCommand(req, maConnexion)

    cmdDelete.Parameters.AddWithValue("@nom", Me.Grid_Entraîneur.Item(0,
Grid_Entraîneur.CurrentRow.Index).Value)

```

```
cmdDelete.Parameters.AddWithValue("@prenom",
Me.Grid_Entraîneur.Item(1, Grid_Entraîneur.CurrentRow.Index).Value)
```

```
cmdDelete.ExecuteNonQuery()
```

```
MsgBox("Entraîneur supprimé...")
```

```
Me.Close()
```

End Sub

Pour supprimer nous utilisons une requête **DELETE**. Elle va nous permettre de supprimer des lignes dans une table. En utilisant cette commande associée à **WHERE** il est possible de sélectionner les lignes concernées qui seront supprimées.

Ensuite, nous utilisons les éléments **cmdDelete. Parameters. AddWithValue** car ils nous permettent de récupérer les résultats de la requête.

Il nous reste plus qu'à exécuter la requête dans la base de données. Nous choisissons d'utiliser le **MsgBox** pour afficher ce message lorsque nous aurons supprimé l'entraîneur.

Puis nous pouvons tester.

Nom de l'entraîneur	Prenom de l'entraîneur
TOURNIE	Alain
BOURGEOIS	Agnès
BOGUSZ	Thierry
TECHER	Charles
PASQUALINI	Claude
Dupont	Jack
TOTO	TOTO
SUN	Christian

Enfin, nous pouvons vérifier dans la base de données si l'entraîneur a bien été supprimé.

ENTRAINEUR				
	NUM_ENTR	NUM_CLUB	NOM_ENTR	PRENOM_ENTR
+	1	1	TOURNIE	Alain
+	2	1	BOURGEOIS	Agnès
+	3	3	BOGUSZ	Thierry
+	4	3	TECHER	Charles
+	5	3	PASQUALINI	Claude
+	6	2	Dupont	Jack
+	123	3	TOTO	TOTO

C'est le cas 😊.

Nous n'avons plus qu'à rajouter la procédure pour fermer le formulaire.

```
Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles  
Cmd_Fermer.Click
```

```
Me.Close()
```

```
End Sub
```

Nous avons décidé de faire un menu en plus, afin de rendre tous ces formulaires plus clairs.

MenuEntraîneur

GESTION DES ENTRAINEURS (MENU)

AJOUTER ENTRAINEUR

MODIFIER / SUPPRIMER
ENTRAINEUR

FERMER

```

Public Class MenuGestionEntraîneur

    Private Sub Cmd_Ajouter_Click(sender As Object, e As EventArgs) Handles
Cmd_Ajouter.Click

        Frm_GestionEntraîneur_AJ.Show()

    End Sub

    Private Sub Cmd_Modifier_Supprimer_Click(sender As Object, e As EventArgs)
Handles Cmd_Modifier_Supprimer.Click

        Frm_GestionEntraîneur_MS.Show()

    End Sub

    Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles
Cmd_Fermer.Click

        Me.Close()

    End Sub

End Class

```

T2.3 Affectation des entraîneurs comme membre du jury

Rappel des objectifs

Cette tâche doit permettre la création, la modification et la suppression de l'affectation d'entraîneurs à une compétition. Là aussi, il faudra proposer une maquette du formulaire au chef de projet. Prévoir tous les contrôles nécessaires au bon fonctionnement de cette tâche.

Formulaire d'affectation des entraîneurs comme membre du jury Création, Modification, Suppression

Comme indiqué dans la consigne nous avons de nouveau préparé nos maquettes en avance afin de les faire valider par les chefs de projet.

1er maquette : Formulaire Ajouter

AFFECTATION DES ENTRAINEURS (AJOUTER)

DateGridView

DateGridView

NUMERO DU JURY :

AJOUTER

FERMER

Après que la première maquette fut validée, nous avons commencé le design et le codage sur **Visual Studio**. Ce dernier est identique à ceux des autres maquettes. Ensuite nous avons rajouté les composants graphiques utiles. Ces derniers ont été renommé comme demandé dans le **Dokuwiki**.

Grid_Entraîneur	Grid_Competition
Txt_Jury	Cmd_Ajouter
Cmd_Fermer	Frm_AffectationEntraîneur_AJ

Aperçu du premier formulaire avant codage

AFFECTATION DES ENTRAINEURS (AJOUTER)

	Numéro de l'entraîneur	Nom de l'entraîneur	Prénom de l'entraîneur
*			

	Numéro de la Compétition	Date de la compétition
*		

NUMERO DU JURY :

AJOUTER

FERMER

Comme pour les autres formulaires, nous avons rempli la **DataGridView**, annoncé la requête puis créé le **DataReader** afin de l'exécuter.

```
Private Sub Frm_AffectationEntraîneur_AJ_Load(sender As Object, e As
EventArgs) Handles MyBase.Load

    connexion("bdKarate.accdb")

    Grid_Entraîneur.ColumnCount = 3

    Grid_Entraîneur.Columns(0).Width = 100

    Grid_Entraîneur.Columns(1).Width = 180

    Grid_Entraîneur.Columns(2).Width = 180

    Grid_Entraîneur.Columns(0).Name = "Numéro"

    Grid_Entraîneur.Columns(1).Name = "Nom"

    Grid_Entraîneur.Columns(2).Name = "Prénom"

    Grid_Entraîneur.AllowUserToAddRows = False

    Dim req As String

    req = "select NUM_ENTRAINEUR, NOM_ENTRAINEUR, PRENOM_ENTRAINEUR from
ENTRAINEUR"

    Dim cmdRech As New OleDbCommand(req, maConnexion)

    Dim rdrRech As OleDbDataReader = cmdRech.ExecuteReader()

    Dim numLigne As Integer

    While (rdrRech.Read())

        Grid_Entraîneur.Rows.Add()

        numLigne = Grid_Entraîneur.RowCount - 1
```

```

        Grid_Entraîneur.Item(0, numLigne).Value =
rdrRech.Item("NUM_ENTRAINEUR")

        Grid_Entraîneur.Item(1, numLigne).Value =
rdrRech.Item("NOM_ENTRAINEUR")

        Grid_Entraîneur.Item(2, numLigne).Value =
rdrRech.Item("PRENOM_ENTRAINEUR")

End While

```

Puisque nous avons deux **DataGridView**, nous devons remplir la deuxième.

```

Grid_Competition.ColumnCount = 2

Grid_Competition.Columns(0).Width = 200

Grid_Competition.Columns(1).Width = 200

Grid_Competition.Columns(0).Name = "Numero de la competition"

Grid_Competition.Columns(1).Name = "Date de la competition"

Grid_Competition.AllowUserToAddRows = False

Grid_Competition.Columns("Numero du jury").Visible = False

```

Nous n'annoncerons pas la requête dans le **Load** car puisque notre requête doit faire afficher seulement les compétitions auxquelles un entraîneur n'a pas déjà été affecté, elle se chargera lorsque nous cliquerons sur la **Grid_Entraîneur**. Nous allons donc remplir la requête et la **DataReader** dans cette **Grid**.

```

Private Sub Grid_Entraîneur_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles Grid_Entraîneur.CellContentClick

    Dim req1 As String

    req1 = "select NUM_COMPETITION, DATE_COMPETITION FROM COMPETITION

        WHERE NUM_COMPETITION NOT IN(select NUM_COMPETITION from JUGE where
NUM_ENTRAINEUR = @numE)"

```

Notre requête récupère le **NUM_COMPETITION** et la **DATE_COMPETITION**. Nous avons mis un **WHERE** car nous voulons récupérer les **NUM_COMPETITION** qui ne sont pas dans notre

« **select JUGE.NUM_COMPETITION from JUGE** » en précisant que nous voulons celle où l'entraîneur n'est pas encore juge. Grâce à l'opérateur logique **NOT IN** nous pouvons faire cela.

```
cmdRech1.Parameters.AddWithValue("@numE", Me.Grid_Entraîneur.Item(0,
Grid_Entraîneur.CurrentRow.Index).Value)
```

Concernant le **DataReader (rdrRech)**, nous ne récupérons que la condition de la sous-requête (**@numE**) car tout dépend du **NUM_ENTRAINEUR**.

```
Dim rdrRech1 As OleDbDataReader = cmdRech1.ExecuteReader()

    Dim numLigne1 As Integer

    Grid_Competition.Rows.Clear()

    While (rdrRech1.Read())

        Grid_Competition.Rows.Add()

        numLigne1 = Grid_Competition.RowCount - 1

        Grid_Competition.Item(0, numLigne1).Value =
rdrRech1.Item("NUM_COMPETITION")

        Grid_Competition.Item(1, numLigne1).Value =
rdrRech1.Item("DATE_COMPETITION")

    End While
```

Pour éviter que lorsque nous cliquons sur la **Grid_Entraîneur** les données du **NUM_COMPETITION** et de la **DATE_COMPETITION** se cumulent, nous avons mis un **Clear** afin que la grille se vide en début du traitement.

Cette requête retourne plusieurs enregistrements, donc comme pour les autres formulaires, nous utilisons une boucle (**While**).

Précédemment sur conseil du professeur, nous avons mis une requête avec l'agrégat **MAX** afin de récupérer le dernier **NUM_JURY**. Après réflexion, nous ne l'avons pas mis car nous pensons que de cette façon, il ne pourra pas être modifié comme nous souhaitons le faire dans le prochain formulaire. De ce fait le **NUM_JURY** se décidera au hasard.

Nous pouvons enfin tester notre formulaire afin de savoir si la **DataGridView** et l'affichage des données fonctionnent.

Frm_AffectationEntraîneur_AJ

AFFECTATION DES ENTRAINEURS (AJOUTER)

	Numéro de l'entraîneur	Nom de l'entraîneur	Prénom de l'entraîneur
	1	TOURNIE	Alain
▶	2	BOURGEOIS	Agnès
	3	BOGUSZ	Thierry
	4	TECHER	Charles
	5	PASQUALINI	Claude
	6	Dupont	Jack
	123	TOTO	TOTO

	Numéro de la Compétition	Date de la compétition
▶	1	25/02/2012 00:00:00
	4	03/05/2012 00:00:00

NUMERO DU JURY :

AJOUTER
FERMER

Nous allons nous occuper maintenant de l'affectation d'un entraîneur à une compétition à l'aide du bouton **AJOUTER**.

```
Dim req As String

req = "insert into JUGE values (@numC, @numE, @numJ)"

Dim cmdInsert As New OleDbCommand(req, maConnexion)

cmdInsert.Parameters.AddWithValue("@numC", Me.Grid_Competition.Item(0,
Grid_Competition.CurrentRow.Index).Value)
```

```

cmdInsert.Parameters.AddWithValue("@numE", Me.Grid_Entraîneur.Item(0,
Grid_Entraîneur.CurrentRow.Index).Value)

cmdInsert.Parameters.AddWithValue("@numJ", Txt_Jury.Text)

cmdInsert.ExecuteNonQuery()

MsgBox("L'entraîneur a bien été affecté à une compétition... ")

```

Avec la requête **insert into**, nous allons insérer le **NUM_COMPETITION** qui vient de la **Grid_Compétition**, le **NUM_ENTRAINEUR** qui vient de la **Grid_Entraîneur** et le **NUM_JURY** qui vient de la **TextBox**.

Nous n'avons plus qu'à exécuter la requête et faire afficher un message, nous indiquant que l'entraîneur a bien été affecté à une compétition.

Puis nous pouvons tester.

The screenshot shows a Windows application window titled "Frm_AffectationEntraîneur_AJ". The main form has a light blue background and a title bar. The title of the form is "AFFECTATION DES ENTRAINEURS (AJOUTER)".

Inside the form, there are two data grids. The first grid lists coaches with columns: "Numéro de l'entraîneur", "Nom de l'entraîneur", and "Prénom de l'entraîneur". The second grid lists competitions with columns: "Numéro de la Compétition" and "Date de la compétition".

Below the grids, there is a text box labeled "NUMERO DU JURY :" with the value "5" entered. At the bottom of the form, there are two buttons: "AJOUTER" and "FERMER".

A message box is overlaid on the form, titled "VB.NET Karaté". The message text is "L'entraîneur a bien été affecté à une compétition...". The message box has an "OK" button.

Numéro de l'entraîneur	Nom de l'entraîneur	Prénom de l'entraîneur
1	TOURNIE	Alain
2	BOURGEOIS	Agnès
3	BOGUSZ	Thieny
4	TECHER	Charles
5	PASQUALINI	Claude
6	Dupont	Jack
123	TOTO	TOTO

Numéro de la Compétition	Date de la compétition
3	10/12/2011 00:00:00
4	03/05/2012 00:00:00

NUMERO DU JURY : 5

AJOUTER FERMER

VB.NET Karaté

L'entraîneur a bien été affecté à une compétition...

OK

Enfin, nous pouvons vérifier dans la base de données si l'entraîneur numéro 4 a bien été affecté à la compétition 3.

JUGE		
NUM_COMP	NUM_ENTR	NUM_JURY
1	1	3
1	3	3
1	4	4
1	5	5
2	1	3
2	2	2
2	3	1
2	4	3
2	5	2
3	2	5
3	4	5
4	3	3

C'est le cas 😊.

Nous n'avons plus qu'à rajouter la procédure pour fermer le formulaire.

```
Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles  
    Cmd_Fermer.Click  
  
        Me.Close()  
  
End Sub
```

Modifier

2ème maquette : Formulaire Modifier/Supprimer

AFFECTATION DES ENTRAINEURS (MODIFIER / SUPPRIMER)

DateGridView

DateGridView

NUMERO DU JURY :

FERMER

MODIFIER

SUPPRIMER

Après que la deuxième maquette fut validée, nous avons rajouté les composants graphiques utiles. Ces derniers ont été renommé comme demandé dans le **Dokuwiki**.

Grid_Entraîneur	Grid_Competition
Txt_Jury	Cmd_Modifier
Cmd_Supprimer	Cmd_Fermer
Frm_AffectationEntraîneur_AJ	

Aperçu du deuxième formulaire avant codage

Le codage concernant les deux **DataGridView** et l'affichage est partiellement identique à celui du formulaire **AJOUTER** pour l'affectation des entraîneurs. En effet, nous modifions seulement la requête.

Dans le formulaire précédent nous voulions faire afficher les compétitions auxquelles un entraîneur n'a pas été affecté. Dans celui-là nous voulons faire afficher l'inverse (les compétitions auxquelles un entraîneur a été affecté). Il nous faut donc une requête avec l'opérateur logique **IN**.

```
Private Sub Grid_Entraîneur_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles Grid_Entraîneur.CellContentClick

    Dim req1 As String

    req1 = "select NUM_COMPETITION, DATE_COMPETITION FROM COMPETITION

    WHERE NUM_COMPETITION IN(select NUM_COMPETITION from JUGE where
NUM_ENTRAINEUR = @numE)"
```

Nous allons maintenant nous occuper de la modification de l'affectation d'un entraineur à une compétition à l'aide du bouton **MODIFIER**.

```
Private Sub Cmd_Modifier_Click(sender As Object, e As EventArgs) Handles
Cmd_Modifier.Click

    Dim req As String

    req = "UPDATE JUGE SET NUM_JURY = @numJ WHERE NUM_COMPETITION = @numC
and NUM_ENTRAINEUR = @numE"

    Dim cmdUpdate As New OleDbCommand(req, maConnexion)

    cmdUpdate.Parameters.AddWithValue("@numJ", Txt_Jury.Text)

    cmdUpdate.Parameters.AddWithValue("@numC", Me.Grid_Competition.Item(0,
Grid_Competition.CurrentRow.Index).Value)

    cmdUpdate.Parameters.AddWithValue("@numE", Me.Grid_Entraîneur.Item(0,
Grid_Entraîneur.CurrentRow.Index).Value)

    cmdUpdate.ExecuteNonQuery()

    MsgBox("L'affectation d'un entraineur à une compétition a bien été
modifiée... ")

    Me.Close()

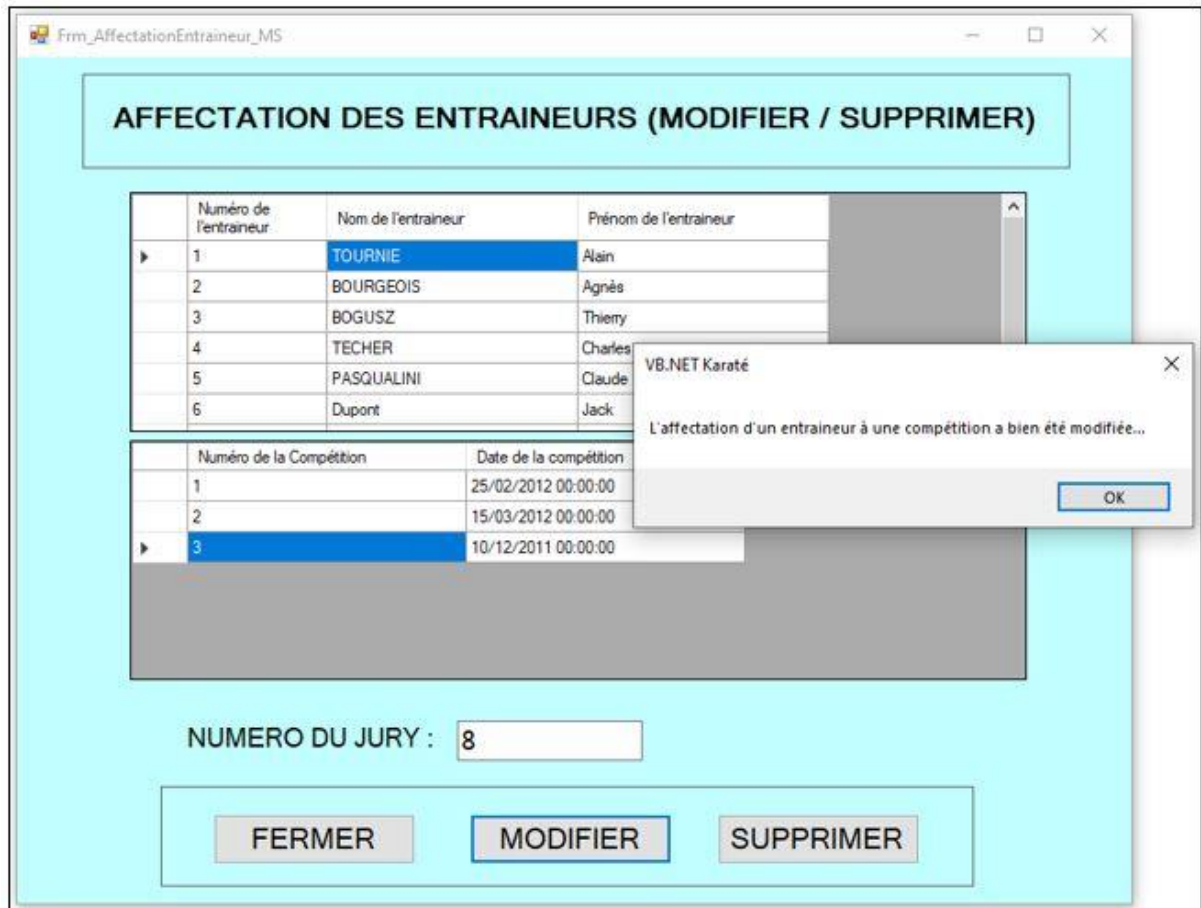
End Sub
```

Pour modifier nous utilisons une requête **UPDTATE**. Elle va nous permettre d'attribuer une nouvelle valeur à une colonne pour les lignes qui respectent la condition stipulé avec **WHERE**.

Nous ne pouvons modifier que le **NUM_JURY**, car le **NUM_COMPETITION** et le **NUM_ENTRAINEUR** sont des clés primaires. De ce fait, nous ne pouvons pas les modifier.

Ensuite, nous récupérerons les résultats de la requête. Il nous reste plus qu'à l'exécuter dans la base de données. Avec le **MsgBox**, nous affichons un message lorsque nous aurons modifier l'affectation d'un entraineur à une compétition.

Puis nous pouvons tester.



A présent, nous pouvons vérifier dans la base de données si le **NUM_JURY** de l'entraîneur numéro 3 a bien été modifié.

JUGE			
NUM_COMP	NUM_ENTR	NUM_JURY	
1	1	3	
1	4	4	
1	5	5	
2	1	3	
2	2	2	
2	3	1	
2	4	3	
2	5	2	
3	1	8	
3	2	5	
3	4	5	
4	3	3	

Enfin, nous allons nous occuper de la suppression de l'affectation d'un entraineur à une compétition à l'aide du bouton **SUPPRIMER**.

```
Private Sub Cmd_Supprimer_Click(sender As Object, e As EventArgs) Handles Cmd_Supprimer.Click

    Dim req As String

    req = "DELETE FROM JUGE WHERE NUM_COMPETITION = @numC and NUM_ENTRAINEUR = @numE"

    Dim cmdDelete As New OleDbCommand(req, maConnexion)

    cmdDelete.Parameters.AddWithValue("@numC", Me.Grid_Competition.Item(0, Grid_Competition.CurrentRow.Index).Value)

    cmdDelete.Parameters.AddWithValue("@numE", Me.Grid_Entraîneur.Item(0, Grid_Entraîneur.CurrentRow.Index).Value)

    cmdDelete.ExecuteNonQuery()

    MsgBox("L'affectation d'un entraineur à une compétition a bien été supprimée... ")

    Me.Close()

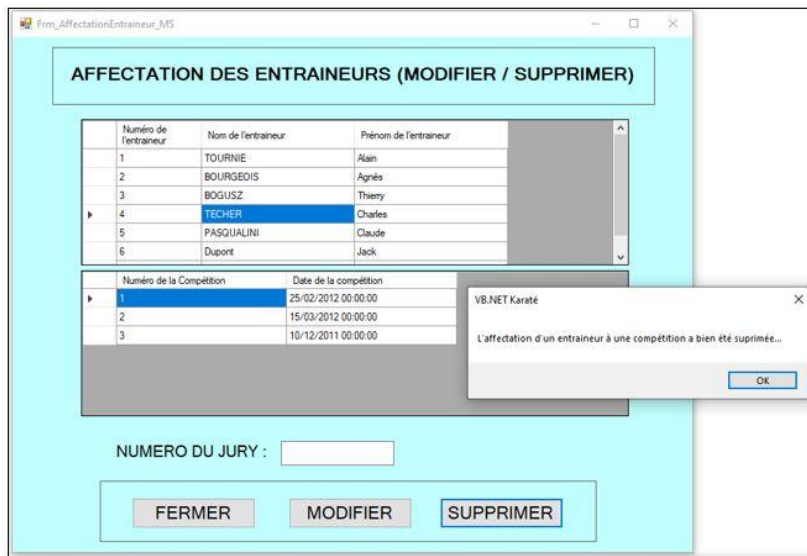
End Sub
```

Pour supprimer nous utilisons une requête **DELETE**. Avec le **WHERE** il est possible de sélectionner les lignes concernées qui seront supprimées. Dans le cas de cette requête, nous supprimons les lignes des champs **NUM_COMPETITION** et **NUM_ENTRAINEUR** de la table **JUGE**. Nous ne mettons pas le **NUM_JURY** car il se supprime automatiquement.

Puis nous récupérons le résultat de la requête.

Nous n'avons plus qu'à exécuter comme précédemment dans le premier formulaire la requête dans la base de données.

Puis nous pouvons tester.



Enfin, nous pouvons vérifier dans la base de données si l'affectation de l'entraîneur numéro 4 à la compétition 1 a bien été supprimé.

JUGE			
NUM_COMP	NUM_ENTR	NUM_JURY	
1	1	3	
#Supprimé	#Supprimé	#Supprimé	
1	4	4	
1	5	5	
2	1	3	
2	2	2	
2	3	1	
2	4	3	
2	5	2	
3	2	5	
3	4	5	
4	3	3	

C'est le cas 😊.

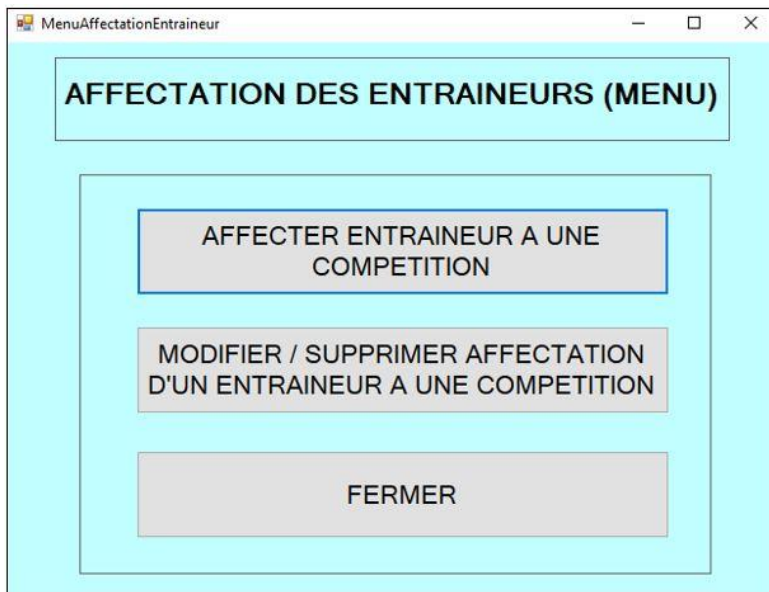
Nous n'avons plus qu'à rajouter la procédure pour fermer le formulaire.

```
Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles  
Cmd_Fermer.Click
```

```
Me.Close()
```

```
End Sub
```

Nous avons décidé de faire un menu en plus, afin de rendre ces deux formulaires plus clairs.



```
Public Class MenuAffectationEntraîneur

    Private Sub Cmd_Ajouter_Click(sender As Object, e As EventArgs) Handles Cmd_Ajouter.Click

        Frm_AffectationEntraîneur_AJ.Show()

    End Sub

    Private Sub Cmd_Modifier_Supprimer_Click(sender As Object, e As EventArgs) Handles Cmd_Modifier_Supprimer.Click

        Frm_AffectationEntraîneur_MS.Show()

    End Sub

    Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles Cmd_Fermer.Click

        Me.Close()

    End Sub

End Class
```

T2.6 Affichage d'un bilan global sur une saison pour un compétiteur donné

Rappel des objectifs

Cette tâche doit permettre l'affichage d'un récapitulatif par saison pour un candidat donné. À la suite de la saisie d'un membre de la Ligue, le document doit faire apparaître dans un tableau la liste des compétitions auxquelles a participé le membre en indiquant pour chacune sa date, son lieu et la note globale obtenue.

Formulaire d'affichage d'un bilan global sur une saison pour un compétiteur donné

Après consultation de la consigne, nous avons décidé de faire deux **DataGridView**.

La première doit afficher le nom et le prénom du membre ainsi que deux **Button** (**AFFICHER** et **FERMER**).

Et la deuxième doit afficher les informations demandées : Liste des compétitions auxquelles a participé le membre, la date, le lieu et la note globale obtenue.

Maquette de premier formulaire avant codage

A screenshot of a Windows form titled 'Frm_Affichage_Global_1'. The form has a light blue background. At the top, there is a title bar with the text 'Frm_Affichage_Global_1' and standard window controls. Below the title bar, there is a rectangular box with a black border containing the text 'AFFICHAGE D'UN BILAN GLOBAL SUR UNE SAISON POUR UN COMPETITEUR DONNÉ'. In the center of the form, there is a larger rectangular box with a black border. Inside this box, at the top, are two text labels: 'Nom du Membre' and 'Prénom du Membre'. Below these labels is a large, empty rectangular area with a gray background, intended for a DataGrid. At the bottom of the form, there are two buttons: 'AFFICHER' on the left and 'FERMER' on the right, both with a light gray background and black text.

Les composants graphiques ont été renommé comme demandé dans le **Dokuwiki**.

Grid_Global_Membre	Cmd_Afficher
Cmd_Fermer	Frm_Affichage_Global

Nous commençons tout d'abord par remplir la **DataGridView**, annoncer notre requête puis créer le **DataReader** afin de l'exécuter.

```
Private Sub Frm_Affichage_Global_Load(sender As Object, e As EventArgs)  
Handles MyBase.Load
```

```
    connexion("bdKarate.accdb")
```

```
    Grid_Global_Membre.ColumnCount = 4
```

```
    Grid_Global_Membre.Columns(0).Width = 100
```

```
    Grid_Global_Membre.Columns(1).Width = 190
```

```
    Grid_Global_Membre.Columns(2).Width = 190
```

```
    Grid_Global_Membre.Columns(3).Width = 100
```

```
    Grid_Global_Membre.Columns(0).Name = "Numéro de licence"
```

```
    Grid_Global_Membre.Columns(1).Name = "Nom du membre"
```

```
    Grid_Global_Membre.Columns(2).Name = "Prénom du membre"
```

```
    Grid_Global_Membre.Columns(3).Name = "Numéro du club"
```

```
    Grid_Global_Membre.AllowUserToAddRows = False
```

```
    Grid_Global_Membre.Columns("Numéro de licence").Visible = False
```

```
    Grid_Global_Membre.Columns("Numéro du club").Visible = False
```

```
    Dim req As String
```

```
    req = "SELECT * from MEMBRE ORDER BY NOM_MEMBRE"
```

```
    Dim cmdRech As New OleDbCommand(req, maConnexion)
```

```
    Dim rdrRech As OleDbDataReader = cmdRech.ExecuteReader()
```

```
    While rdrRech.Read()
```

```
        Grid_Global_Membre.Rows.Add()
```

```

        Dim numLigne = Grid_Global_Membre.RowCount - 1

        Grid_Global_Membre.Item(0, numLigne).Value =
rdrRech.Item("NUM_LICENCE")

        Grid_Global_Membre.Item(1, numLigne).Value =
rdrRech.Item("NOM_MEMBRE")

        Grid_Global_Membre.Item(2, numLigne).Value =
rdrRech.Item("PRENOM_MEMBRE")

        Grid_Global_Membre.Item(3, numLigne).Value =
rdrRech.Item("NUM_CLUB")

    End While

End Sub

```

Nous avons caché les colonnes **NUM_LICENCE** et **NUM_CLUB** car ce n'est pas utile qu'elles soient affichées.

```

Private Sub Grid_Global_Membre_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles Grid_Global_Membre.CellContentClick

        idMembre = Me.Grid_Global_Membre.Item(0,
Me.Grid_Global_Membre.CurrentRow.Index).Value

        Id_Membre = idMembre

End Sub

```

Dans la **Grid** nous mettons la variable **idMembre** qui correspond au **NUM_LICENCE**. En effet, c'est ce dernier qui va nous permettre de ne faire afficher que les informations qui correspondent aux membres sélectionnés.

```

Private Sub Cmd_Afficher_Click(sender As Object, e As EventArgs) Handles
Cmd_Afficher.Click

        Frm_Affichage_Global_Resultat.Show()

End Sub

```

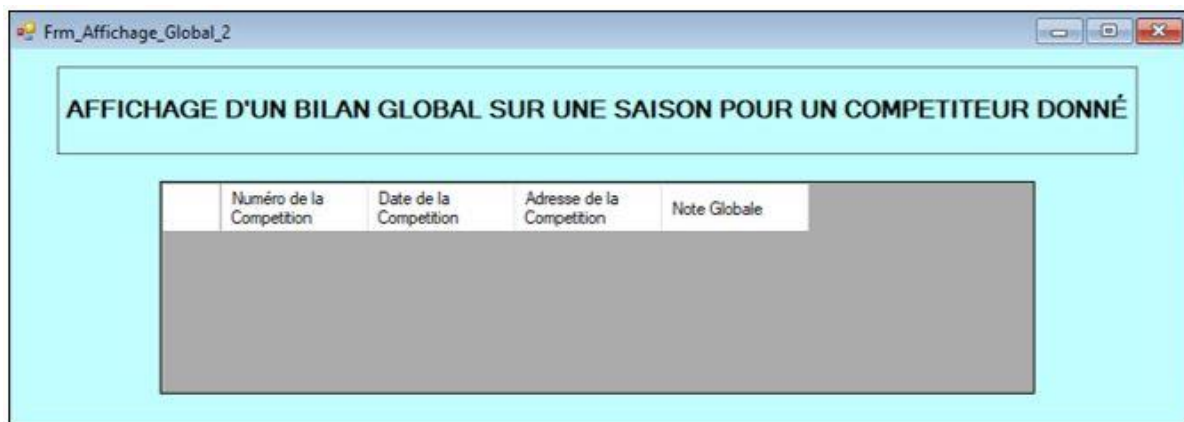
Comme nous l'avons dit précédemment, nous souhaitons que lorsque nous cliquons sur le bouton **AFFICHER**, un nouveau formulaire s'ouvre avec les informations. Pour cela nous faisons un lien avec l'autre formulaire que nous avons nommé **Frm_Affichage_Global_Resultat**. C'est avec la méthode **Show** que nous faisons cela.

Puis nous fermons le formulaire avec le **Close()**.

```
Private Sub Cmd_Fermer_Click(sender As Object, e As EventArgs) Handles  
    Cmd_Fermer.Click  
  
        Close()  
  
End Sub
```

A présent nous pouvons passer au deuxième formulaire.

Maquette de deuxième formulaire avant codage



Les composants graphiques ont été renommé comme demandé dans le **Dokuwiki**.

Grid_Résultat_Bilan	Frm_Affichage_Global_Resultat
---------------------	-------------------------------

Nous commençons tout d'abord par remplir la **DataGridView**.

```
Private Sub Frm_Affichage_Global_2_Load(sender As Object, e As EventArgs)  
    Handles MyBase.Load  
  
        Grid_Résultat_Bilan.ColumnCount = 4  
  
        Grid_Résultat_Bilan.Columns(0).Width = 100  
  
        Grid_Résultat_Bilan.Columns(1).Width = 150  
  
        Grid_Résultat_Bilan.Columns(2).Width = 200  
  
        Grid_Résultat_Bilan.Columns(3).Width = 100  
  
        Grid_Résultat_Bilan.Columns(0).Name = "Numéro de la compétition"
```



```

Grid_Résultat_Bilan.Columns(1).Name = "Date de la compétition"

Grid_Résultat_Bilan.Columns(2).Name = "Adresse, ville et code postal
du Club"

Grid_Résultat_Bilan.Columns(3).Name = "Note globale"

```

Puis nous annonçons la requête.

```

Dim req As String

req = "SELECT INSCRIPTION.NUM_COMPETITION, DATE_COMPETITION,
ADR_RUE_CLUB, CODE_POST_CLUB, ADR_VILLE_CLUB, NOTE_GLOBALE

      FROM (CLUB INNER JOIN COMPETITION ON CLUB.NUM_CLUB =
COMPETITION.NUM_CLUB)INNER JOIN INSCRIPTION ON COMPETITION.NUM_COMPETITION =
INSCRIPTION.NUM_COMPETITION

      WHERE (((INSCRIPTION.NUM_LICENCE)=[@id]))

      ORDER BY INSCRIPTION.NUM_COMPETITION"

```

Notre requête récupère le **NUM_COMPETITION**, la **DATE-COMPETITION**, l'**ADR_RUE_CLUB**, le **CODE_POST_CLUB**, l'**ADR_VILLE_CLUB** et la **NOTE_GLOBALE**. Comme nous sélectionnons plusieurs champs, nous devons faire deux jointures. Une entre les tables **CLUB** et **COMPETITION** et une autre entre les tables **COMPETITION** et **INSCRIPTION**. Nous utilisons la condition **WHERE** afin de récupérer les données en fonction du **NUM_LICENCE**. Puis nous mettons un **ORDER BY** afin de trier les champs par **NUM_COMPETITION**.

```
cmdRech.Parameters.AddWithValue("@id", Id_Membre)
```

Concernant le **DataReader (rdrRech)**, nous ne récupérons que la condition de la requête (**@id**) car tout dépend de **NUM_LICENCE**.

```

While rdrRech.Read()

    Grid_Résultat_Bilan.Rows.Add()

    Dim numLigne = Grid_Résultat_Bilan.RowCount - 1

    Grid_Résultat_Bilan.Item(0, numLigne).Value =
rdrRech.Item("NUM_COMPETITION")

    Grid_Résultat_Bilan.Item(1, numLigne).Value =
rdrRech.Item("DATE_COMPETITION")

```

```

Grid_Résultat_Bilan.Item(2, numLigne).Value =
rdrRech.Item("ADR_RUE_CLUB") + " " + rdrRech.Item("CODE_POST_CLUB") + " " +
rdrRech.Item("ADR_VILLE_CLUB")

Grid_Résultat_Bilan.Item(3, numLigne).Value =
rdrRech.Item("NOTE_GLOBALE")

End While

```

Cette requête retourne plusieurs enregistrements, donc comme pour les autres formulaires, nous utilisons une boucle (**While**).

Nous pouvons enfin tester notre formulaire afin de savoir si les **DataGridView** et l'affichage des données fonctionnent.

	Prenom du Membre	Prénom du membre
▶	ASSIN	Marc
	AVOILE	Richard
	MENVUSA	Gérard
	NISCOTCH	Nicole
	SALEE	Clémentine

Nous choisissons le membre Gérard MENVUSA puis nous cliquons sur le bouton **AFFICHER**.

	Numéro de la Competition	Date de la Competition	Adresse de la Competition	Note Globale
▶	1	25/02/2012 00:00:00	45 rue Emile bertin 54000 Nancy	24
	2	15/03/2012 00:00:00	rue de Remsing 57600 Forbach	
	3	10/12/2011 00:00:00	11 rue du Dauphiné 57070 Metz	
	4	03/05/2012 00:00:00	11 rue des Vosges 57070 Metz	

L'affichage d'un bilan global sur une saison pour le compétiteur Gérard MENVUSA a fonctionné 😊.