

Guide de la connexion entre VB.net et Microsoft Access 2003

Introduction :

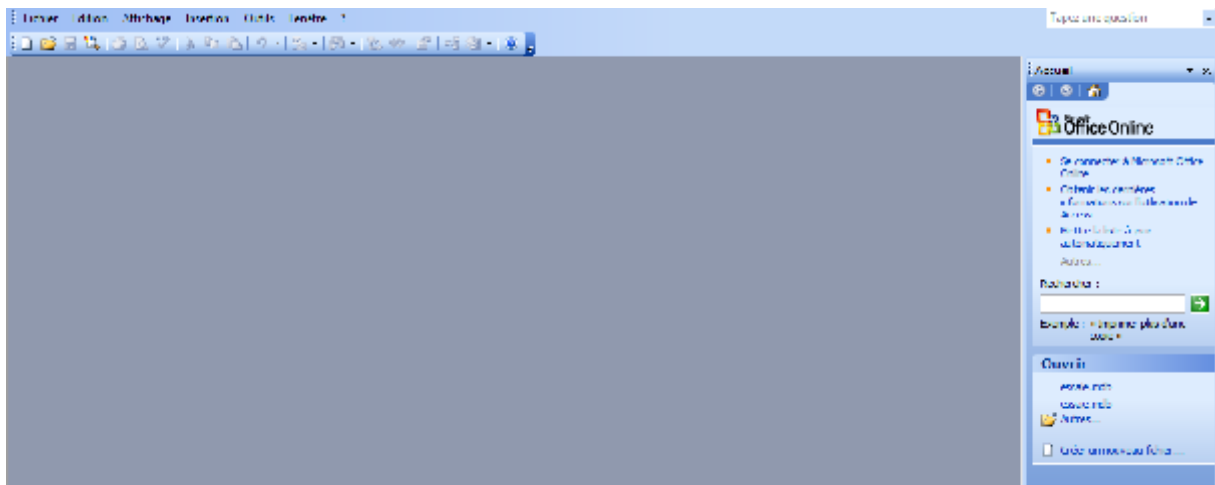
Visual Basic.Net apporte une puissance inégalée et nécessite une rigueur importante mais il devient vite complexe et technique. La documentation et les livres sont totalement hermétiques pour les novices et rebutent totalement les débutants. Les articles sur le Web sont très techniques et traitent d'emblée de problèmes complexes, ils sont nécessaires mais pas pour le débutant. J'explique donc dans ce petit guide, à ma manière, très simplement, comment créer un programme vb.net en le connectant à MS Access afin de permettre un bon démarrage même à celui qui ne l'a jamais fait.

Ms Access :

Avant de pouvoir utiliser une base de donnée créer en Access dans vb.net en utilisant l'IDE Visual studio, nous devons d'abord être en mesure de créer la base, donc la base de donnée (tables, requêtes) dans Access. C'est après la création de cette dernière que nous pouvons être en mesure de la connecter en vb.net en utilisant l'objet OLEDB.

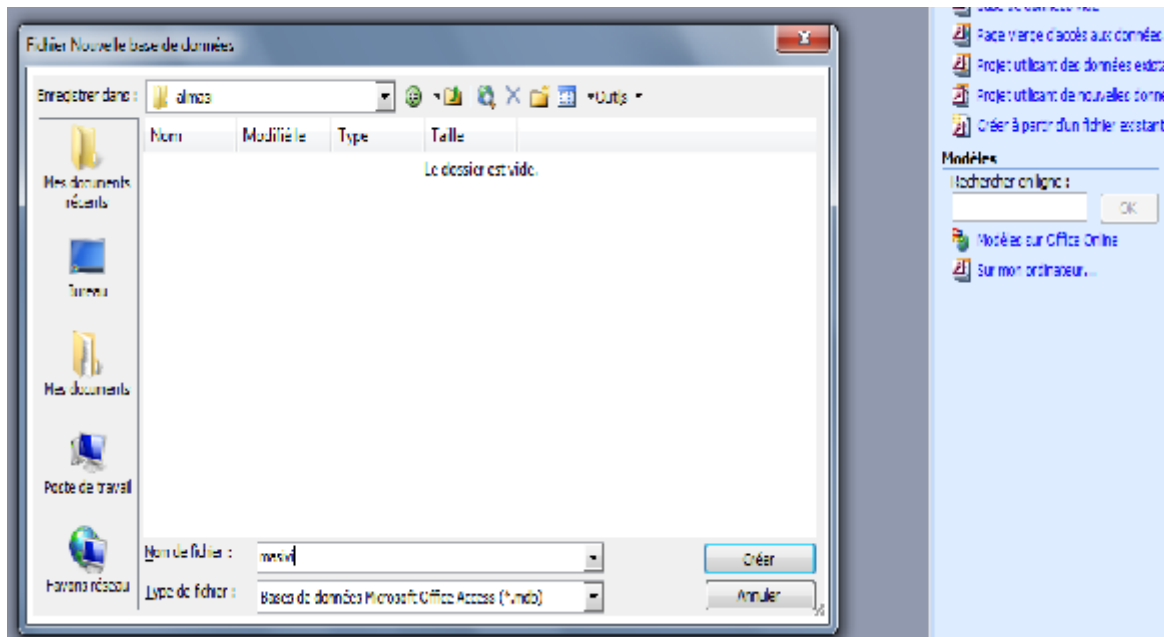
Création de la table :

Pour créer une table en Access, nous ouvrons Ms Access avec un clic de la souris, et nous avons cette fenêtre ci-dessous

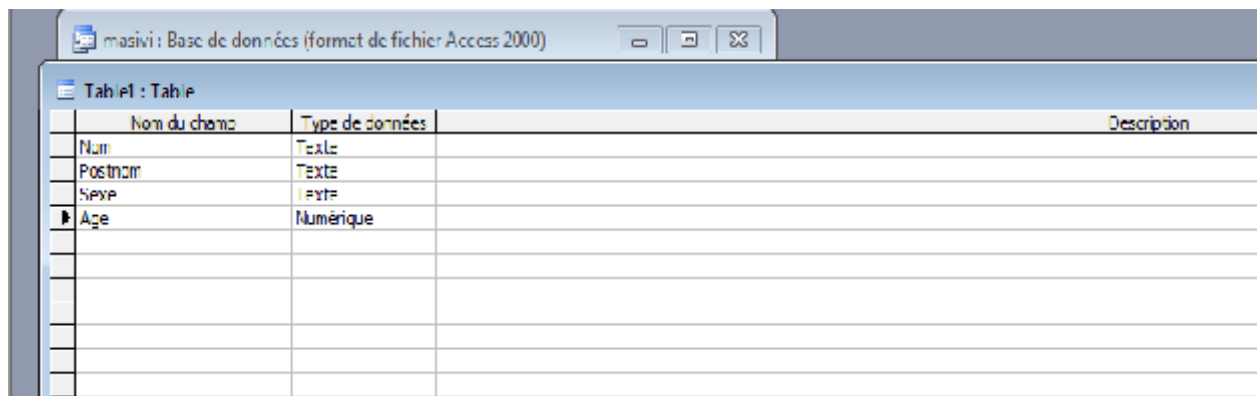


Et puis juste à droite nous y voyons là où on écrit « créer un nouveau fichier » et c'est à partir de là que nous allons créer notre base de donnée. Nous complétons le nom de notre nouvelle base de donnée, puis nous parcourons le répertoire où sera stocké notre base de donnée.

N.B : il faut très bien retenir le répertoire qui stockera le nom de notre base de données. Dans notre exemple le nom de la base de donnée est « **Masivi** » son répertoire est « **C:alimasi/masivi.mdb** »



Et puis nous allons y enregistrer notre base de données. Créons maintenant notre table que nous la renommerons « **etudiant** » avec 4 champs :-nom, post nom, sexe, âge.

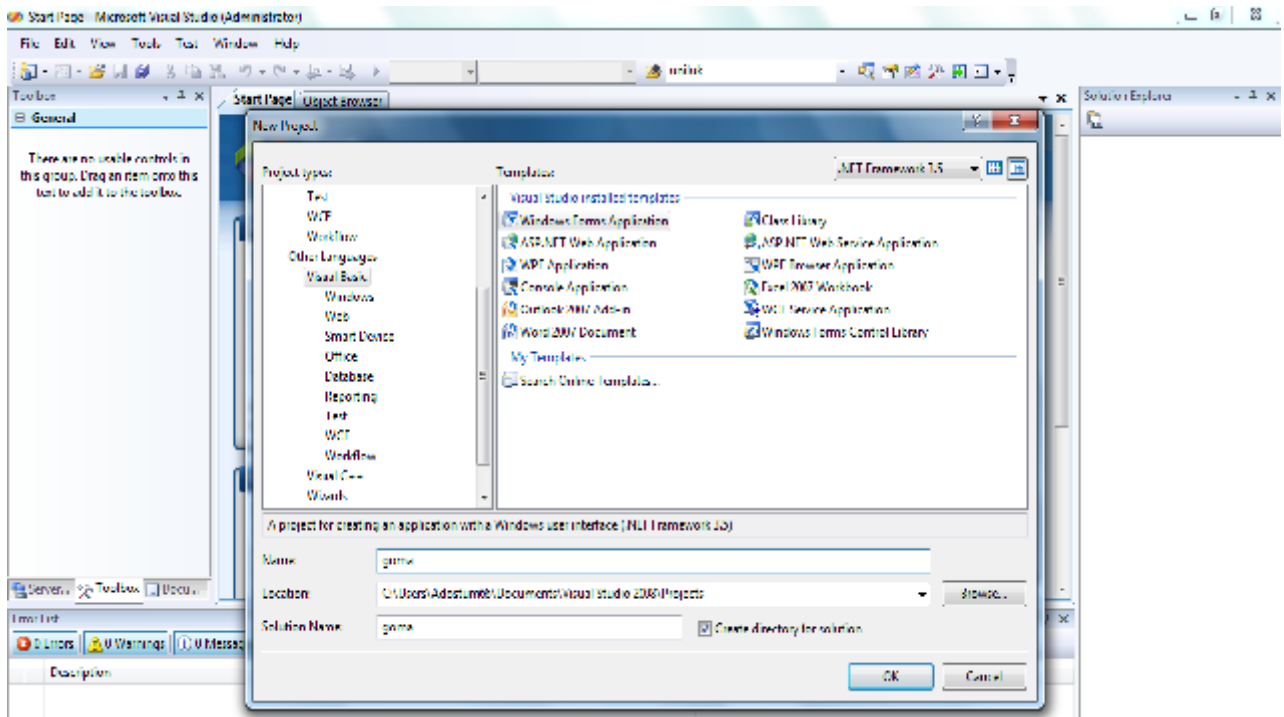


Après la création de notre base de données, vient maintenant le bon moment de sa connexion dans Visual studio en utilisant vb.net

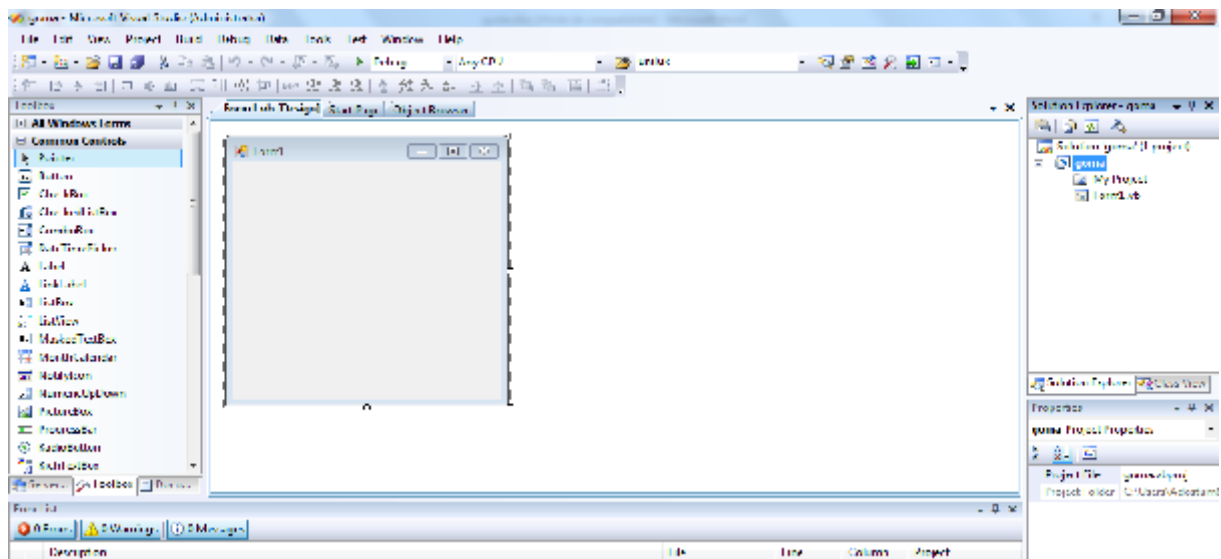
Ouvrons Visual studio et choisissons le langage vb.net en cliquant sur :

- file(Fichier)
- New(Nouveau)
- Project (projet) et voilà que la fenêtre ci-dessous apparaîtra

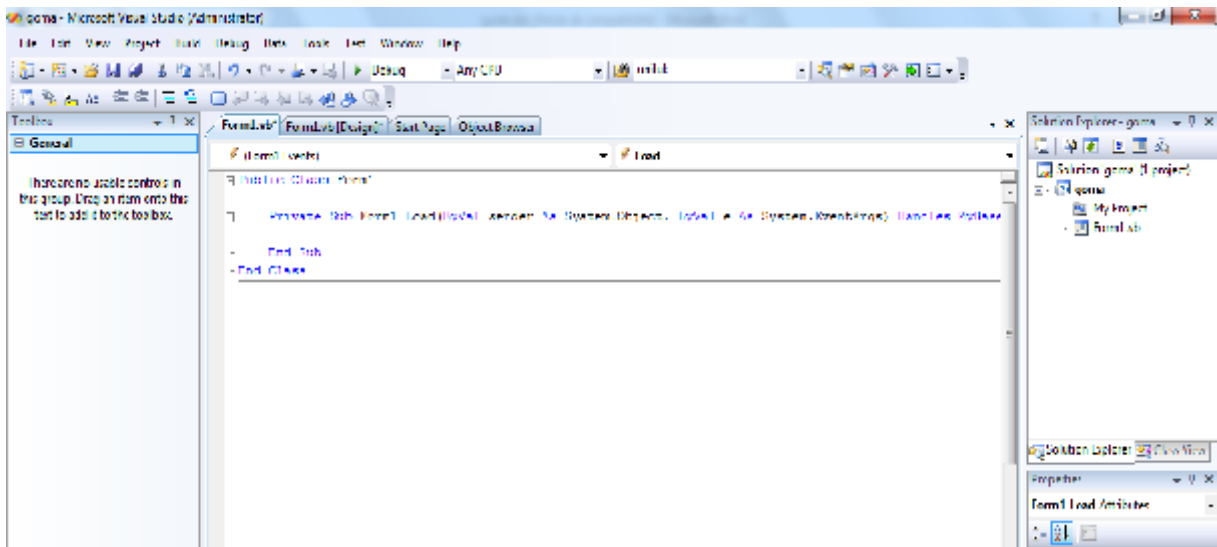
Puis Nous choisirons Visual basic et puis Windows forms application (pour la création d'une application graphique) et dans notre exemple nous renommerons le projet « **Goma** »



Et après cliquons sur le bouton Ok, puis voici notre nouvelle interface de travail



Doublons cliquons sur notre fenêtre pour enfin créer notre connexion.



Pour avoir accès aux données il faut charger les **DRIVERS** (ou providers).

Comme d'habitude, il faut:

- Charger les références des drivers (les Dll)
- Importer les espaces de nom.

Ainsi on a accès aux objets ADO.NET correspondant. Voyons cela:

-**OLE DB** Managed Provider est fourni dans 'System'; après avoir importé le Namespace System.Data.OLEDB, on peut travailler sur des bases Access par exemple.

-**SQL Server** Managed Provider est fourni dans 'System'; après avoir importé le Namespace System.Data.SqlClient, on peut travailler sur des bases SqlServeur.

-Un composant **ODBC** et un composant **ORACLE** sont disponible sur le site MSDN, il faudra charger la référence de la Dll puis le Namespace.

Exemple, pour travailler sur une base Access, il faudra taper:

`Imports System.Data.OLEDB` et c'est celui-ci que nous allons l'utiliser dans notre exemple. Donc avant notre public classe form1 au-dessus nous écrirons

`Imports System.Data.OLEDB` comme c'est lui le driver que nous avons besoin.

Il faut disposer d'un objet **Connexion** pour avoir accès à la base de données, on met dans la propriété

ConnectionString les paramètres de la base de données (nom de la base de données, chemin, mot de passe..).

Déclarons objet qui contiendra la connexion et comme nous utilisons le driver OLEDB, nous allons le déclarer en utilisant [OleDbConnection](#)

Dim con **As New** OleDbConnection, voila la variable de notre connexion

Puis après tout ceci nous aurons besoin d'un datadapteur et d'un dataset. Le **DataSet** peut lire et écrire des données dans la BD, il faut un DataAdapter en plus de la connexion.

Le **DataSet** a la structure d'une base de données mais en local; il contient: Des **DataTable** qui contiennent des **DataRow** et des **DataColumn**. Pour utiliser DataSet, DataTable, DataRow. Il faut importer l'espace de nom Data : donc nous allons aussi au-dessus de notre classe écrire

Imports System.Data

Pour enfin avoir ces lignes de codes :

Imports System.Data.OleDb

Imports System.Data

Public Class Form1

Dim con **As New** OleDbConnection

Dim forest **As String**(pour contenir notre instruction sql)

Dim ds **As New** DataSet

Dim da **As** OleDbDataAdapter

Dim maxrows, x **As Integer**(maxrows pour contenir les nombres totaux de nos lignes, et la variable x contenir une ligne)

Private Sub Form1_Load(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MyBase.Load

con.ConnectionString = ("Provider=Microsoft.Jet.OLEDB.4.0;Data source=C:/alimasi/masivi.mdb")

con.Open()

forest = "select * from etudiant"

da = **New** OleDb.OleDbDataAdapter(forest, con)

da.Fill(ds, "masivi")

con.Close()

maxrows = ds.Tables("masivi").Rows.Count

x = 0

```

    osee()
End Sub

```

Pour afficher les données de notre BD pendant le lancement de notre application, créons une procédure et affectons les données de nos tables aux zones de textes

```

Private Sub osee()
    'fonction pour affectation données dans nos zones de textes
    TextBox1.Text = ds.Tables("masivi").Rows(x).Item(0)
    TextBox2.Text = ds.Tables("masivi").Rows(x).Item(1)
    TextBox3.Text = ds.Tables("masivi").Rows(x).Item(3)
    If ds.Tables("masivi").Rows(x).Item(2) = "M" Then
        RadioButton1.Checked = True
    Else
        RadioButton2.Checked = True
    End If
End Sub

```

Et voici le code au complet

```

Imports System.Data.OleDb
Imports System.Data
Public Class Form1
    Dim con As New OleDbConnection
    Dim forest As String
    Dim ds As New DataSet
    Dim da As OleDbDataAdapter
    Dim maxrows, x As Integer
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'declaration de la variable de la connexion
        con.ConnectionString = ("Provider=Microsoft.Jet.OLEDB.4.0;Data
source=C:/alimasi/masivi.mdb")
        'ouverture de la connexion
        con.Open()
        'selections des données
        forest = "select * from etudiant"
        'mettre les données dans le dataadaptateur

```

```

da = New OleDb.OleDbDataAdapter(forest, con)
'creation d'un lien entre la bd et le dataset
da.Fill(ds, "masivi")
'fermeture de la connexion
con.Close()
'selection de toute nos lignes
maxrows = ds.Tables("masivi").Rows.Count
'initialisation de nos lignes
x = 0
'appel de notre procedure pour afficher les données
osee()
End Sub
Private Sub osee()
'fonction pour affectation données dans nos zones de textes
TextBox1.Text = ds.Tables("masivi").Rows(x).Item(0)
TextBox2.Text = ds.Tables("masivi").Rows(x).Item(1)
TextBox3.Text = ds.Tables("masivi").Rows(x).Item(3)
If ds.Tables("masivi").Rows(x).Item(2) = "M" Then
    RadioButton1.Checked = True
Else
    RadioButton2.Checked = True
End If
End Sub
End Class

```

INSTRUCTIONS SQL :

Passons maintenant en revue des instructions que nous utiliserons :

ENREGISTREMENT DES DONNES : pour enregistres, nous allons cliquer sur notre bouton qui s'occupera de l'enregistrement pour y mettre nos codes pour introduire les donnes à partir du formulaire, mais avant d'enregistrer les données nous devons rendre toute les zones à vide.

Nous aurons d'abord à initialiser les zones des textes. Cliquons sur le bouton nouveau et mettons y ce code, donc affectons aux zones de text des vides.

```

'initialisation des zones de textes
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
RadioButton1.Checked = False

```

RadioButton2.Checked =False

Puis mettons maintenant dans le bouton enregistrer les codes qui nous aiderons à enregistrer les données.

Try

```

' verification s'il n'ya pas des champs vide
If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Then
    MsgBox("verifiez les champs vide")
Else
    ' ouverture de la connexion
    con.Open()
    'declarons un object command qui va modifier la base de donnee
    Dim ajouter As New OleDbCommand
    'code sql pour l'enregistrement des donnes
    ajouter.CommandText = "insert into etudiant values (" & TextBox1.Text & "
    ',' & TextBox2.Text & ',' & m & ',' & TextBox3.Text & ")"
    ajouter.Connection = con
    ajouter.CommandType = CommandType.Text
    ajouter.ExecuteNonQuery()
    MsgBox("L'enregistrement a été ajouté avec succès")
    con.Close()
End If
Catch ex As Exception
    MsgBox("Erreur System")
End Try

```

SUPPRESSION DES DONNES : à l'aide toujours de notre objet command en complémentarité avec l'instruction sql de la suppression, ajoutons du code dans le bouton delete.

```

'ouverture de la connexion
con.Open()
'declaration de la commande
Dim delete As New OleDbCommand
'declaration d'une variable qui contiendra notre inputbox
Dim k As String
k = InputBox("Entrez le Numero de l'etudiant a supprimer")
If k <> "" Then
    delete.CommandText = "delete from etudiant where Nom=" & k & ""
    delete.Connection = con

```



```

delete.CommandType = CommandType.Text
delete.ExecuteNonQuery()
MsgBox("Un etudiant a ete supprimer")
Else
MsgBox("Rien a supprimer")
End If
con.Close()

```

Rechercher des donnees : en utilisant l'objet command avec l'instruction select avec une clause, créons le bouton rechercher

N.B : Un objet **DataReader** fournit des données en lecture seule en un temps record. La seule possibilité est de se déplacer en avant. **Il faut créer un objet Connexion puis un objet Command**, ensuite on exécute la propriété **ExecuteReader** pour créer l'objet **DataReader**; enfin on parcourt les enregistrements avec la méthode **Read**. Et c'est cet objet que nous allons l'utiliser ici

'ouverture de notre connexion

```
con.Open()
```

'declaration d'un string pour contenir notre inputbox et d'un objet command

```
Dim rechercher As New OleDbCommand
```

```
Dim p As String
```

```
p = InputBox("Entrez le nom de l'etudiant ")
```

```
rechercher.CommandText = "select * from etudiant where Nom=" & p & ""
```

```
rechercher.Connection = con
```

'declaration d'un objet datareader qui nous permettra de lire les donnees de la bd

```
Dim dbreader As OleDbDataReader = rechercher.ExecuteReader
```

```
If dbreader.Read = True Then
```

```
TextBox1.Text = dbreader.Item(0)
```

```
TextBox2.Text = dbreader.Item(1)
```

```
If dbreader.Item(2) = "M" Then
```

```
RadioButton1.Checked = True
```

```
Else
```

```
RadioButton2.Checked = True
```

```
End If
```

```
    TextBox3.Text = dbreader.Item(3)
End If
con.Close()
```

Parcours de la Base de données

Défiler les données : pour défiler les données nous aurons à faire à 4 choses :

Défiler avec le suivant, précédent, premier, dernier.

Suivant enregistrement:

```
    If x < maxrows - 1 Then
        x = x + 1
        'notre procedure
        osee()
    Else
        MsgBox("Vous êtes au dernier enregistrement")
    End If
```

Précédent enregistrement:

```
    If x > 0 Then
        x = x - 1
        'notre procedure
        osee()
    Else
        MsgBox("Vous êtes au premier enregistrement")
    End If
```

Dernier Enregistrement :

```
    If x <> maxrows - 1 Then
        x = maxrows - 1
```

```

osee()
Else
  MsgBox("Vous etes deja au dernier enregistrement")
End If

```

Premier enregistrement :

```

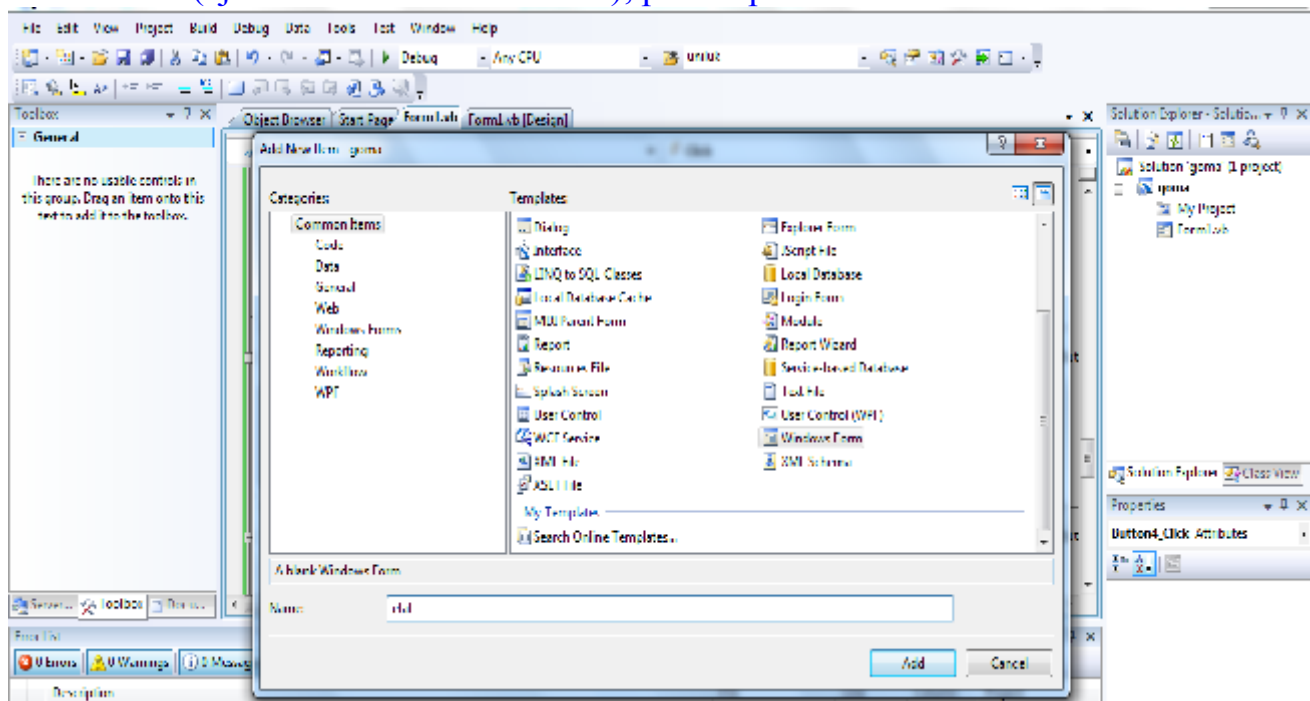
If x <> 0 Then
  x = 0
  osee()
Else
  MsgBox("Vous etes deja au premier enregistrement")
End If

```

CREATION DE CRYSTALL REPORT :

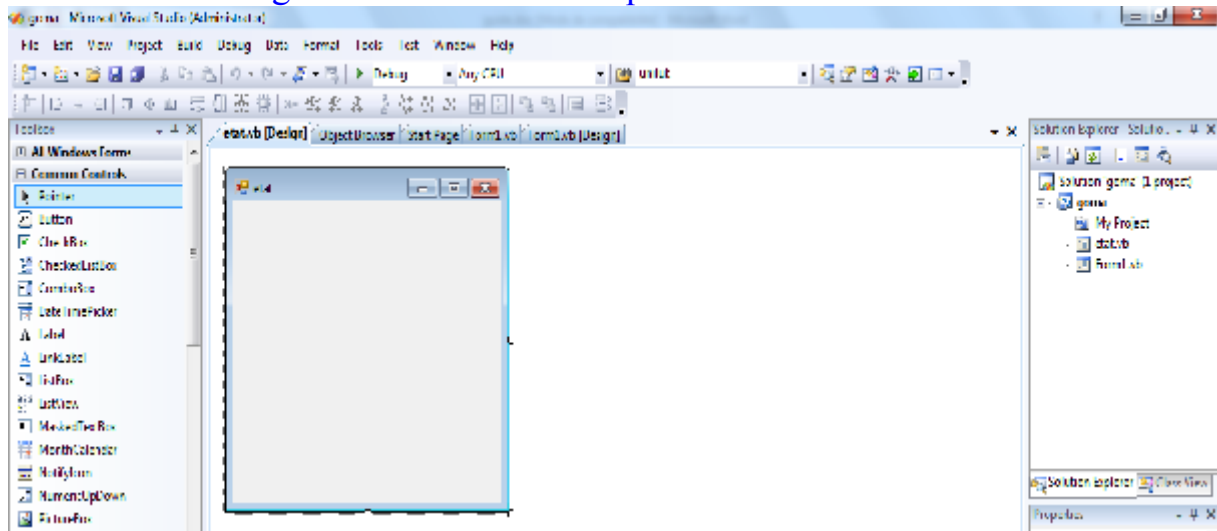
Le crystall report qui nous aidera à visualiser nos données dans le but des les imprimés par exemple. Pour y arrivé nous devons d’abord créer un formulaires qui nous aidera à afficher nos états.

- 1) Ajout du formulaire : allons sur le menus Projet, puis add new windows forms(ajout de la nouvelle fenetre), puis baptisons le au nom « etat »

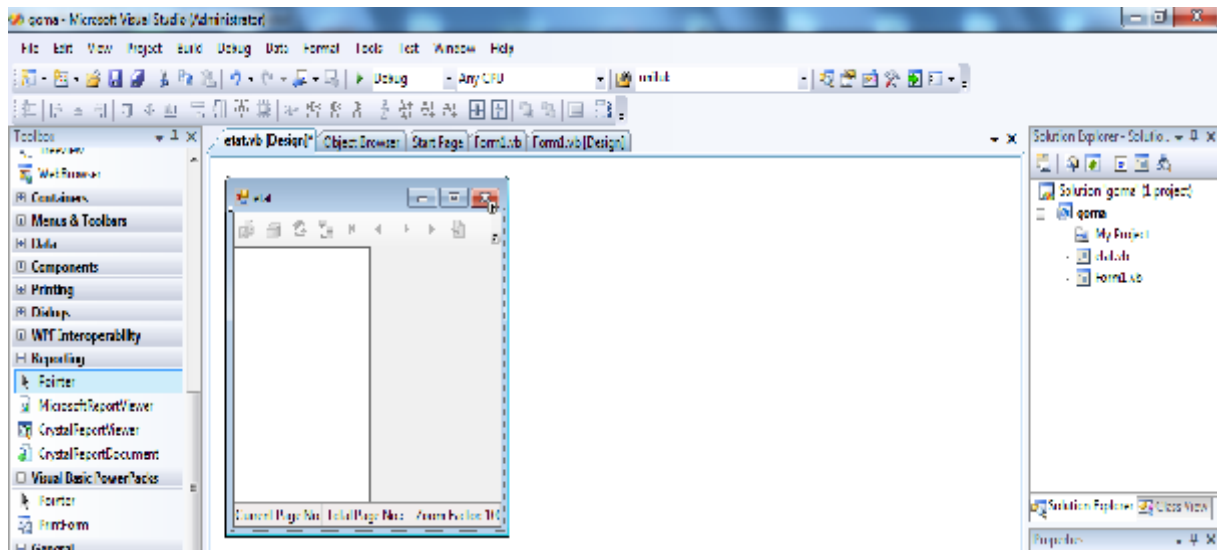


Puis cliquons sur le bouton add.

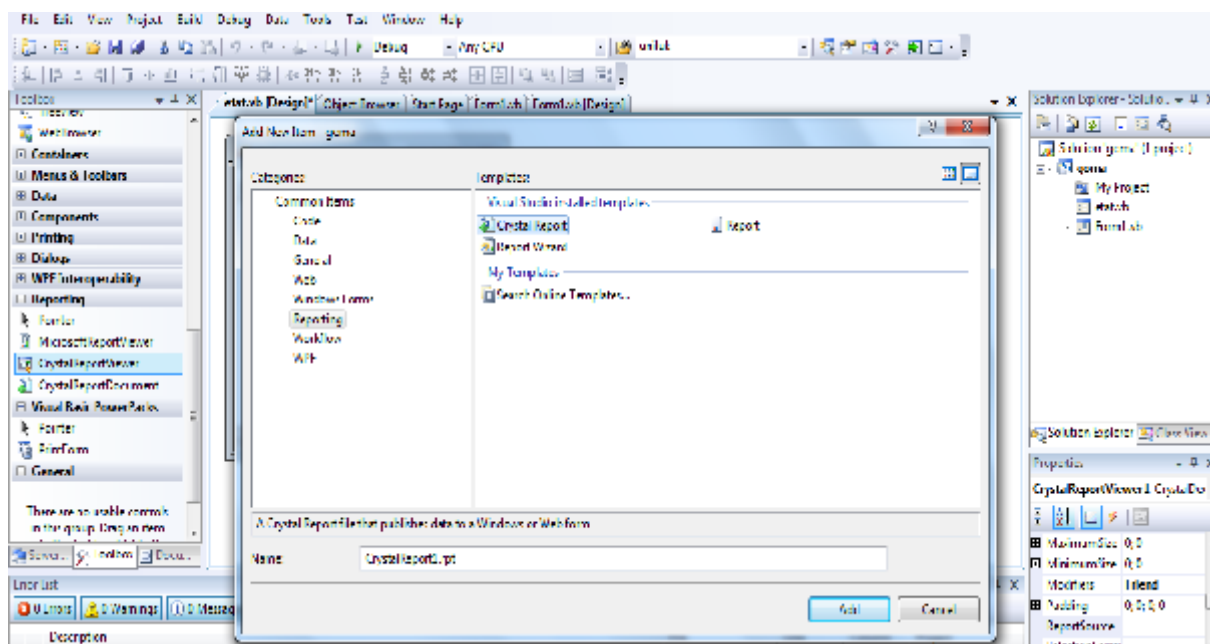
2. ouvrons le design de notre formulaire que nous venons de créer.



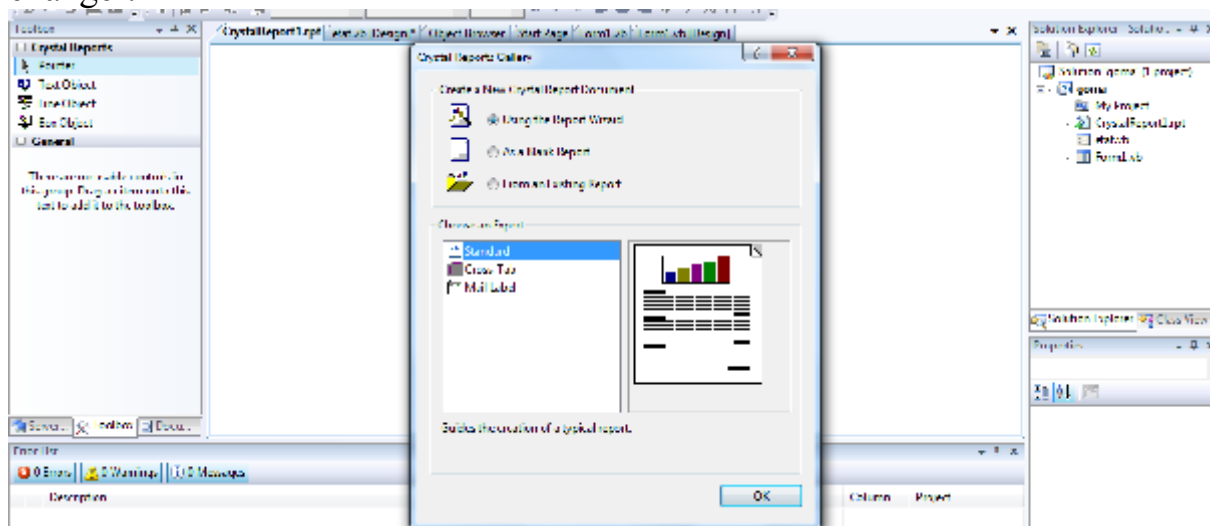
3. ajoutons sur notre formulaire un contrôle de la boîte d'outils dans la partie Reporting qui s'appelle « CrystalReportViewer »



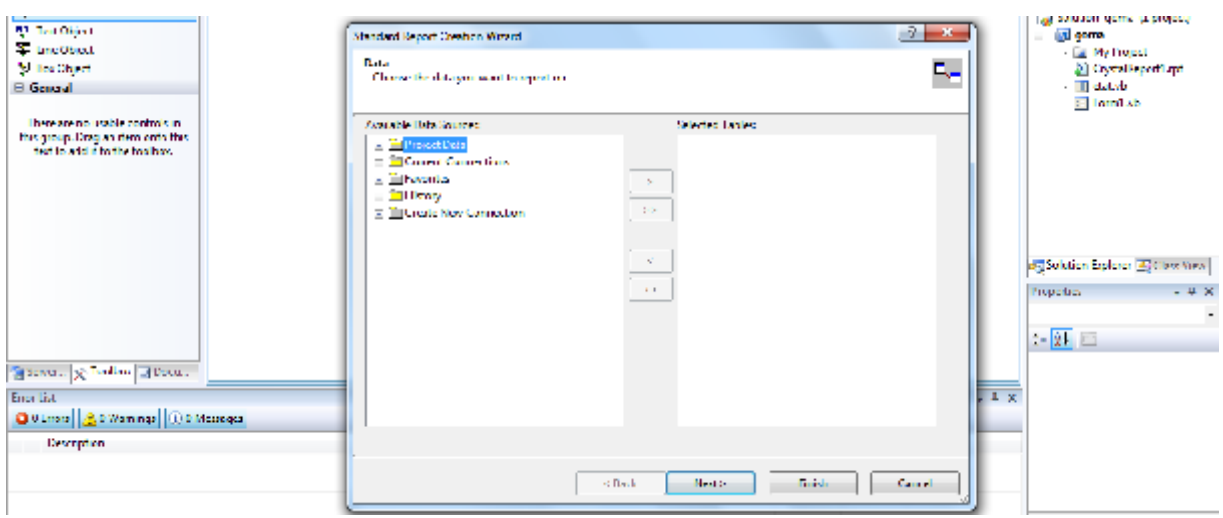
4. créons maintenant notre crystalreport, en cliquant sur le menu, projet, new windows form, puis sous menu reporting, et sélectionnons Crystal Report



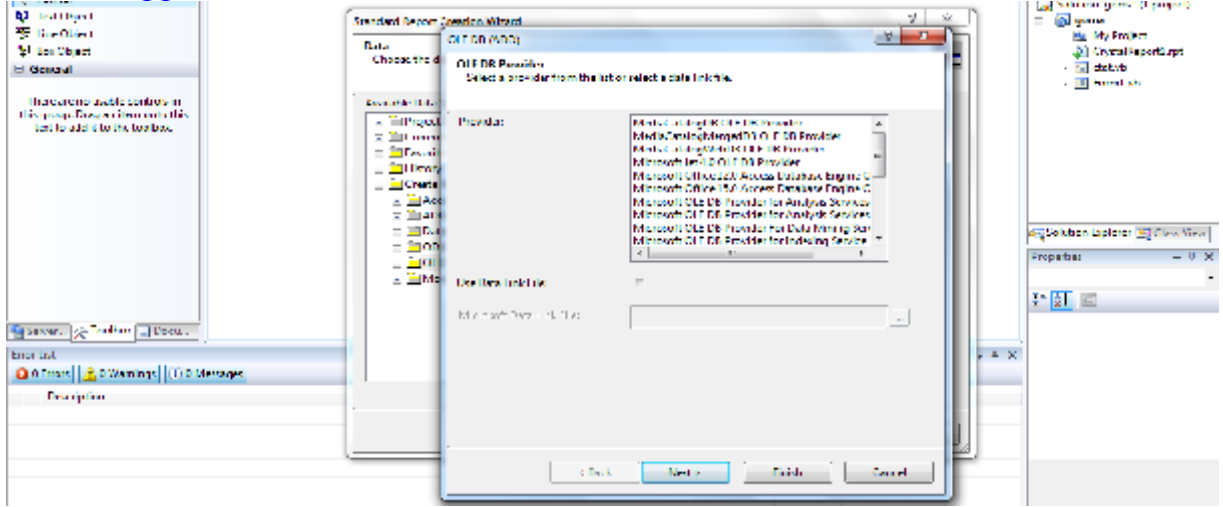
Après quelques secondes une nouvelle fenetre apparait cliquons sur ok sans rien changer.



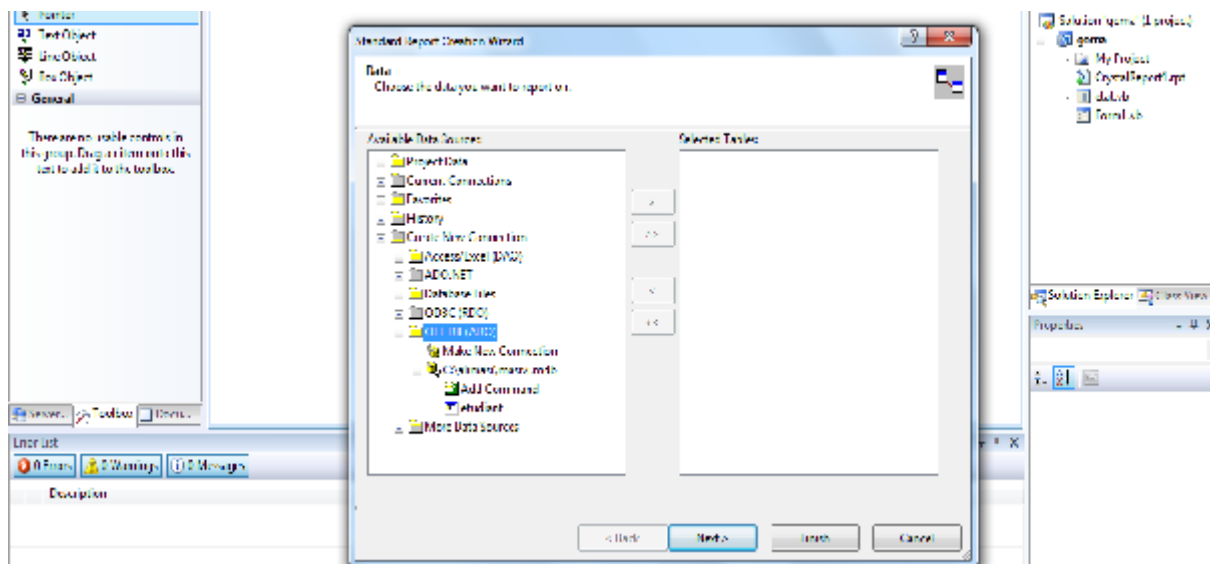
Une nouvelle fenetre apparait qui nous obligera à selectionner la source de notre base de donnee.



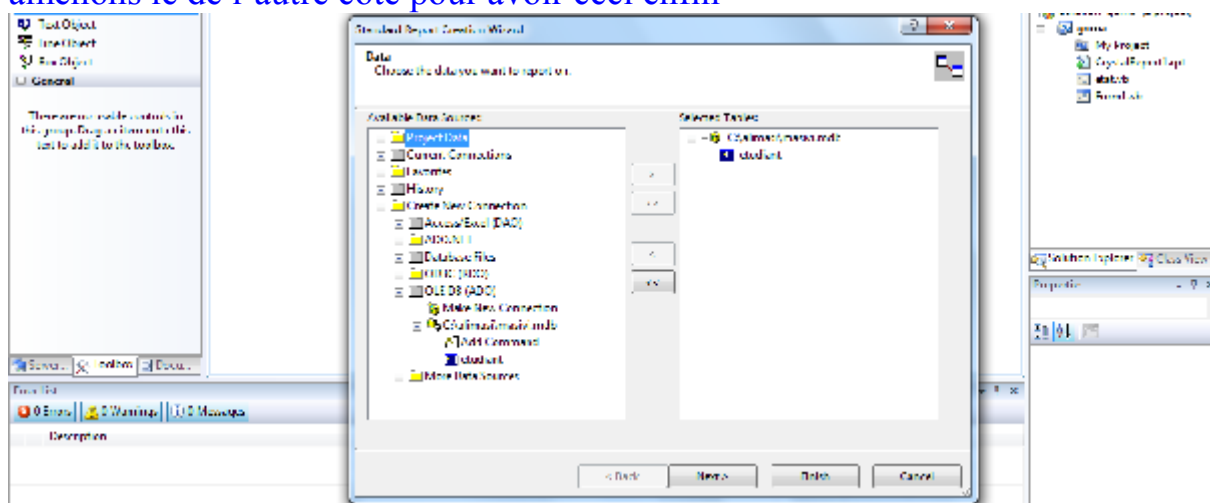
Cliquons sur create new connection, puis OLEDB(ADO), et une fenetre comme celle-ci apparaîtra.



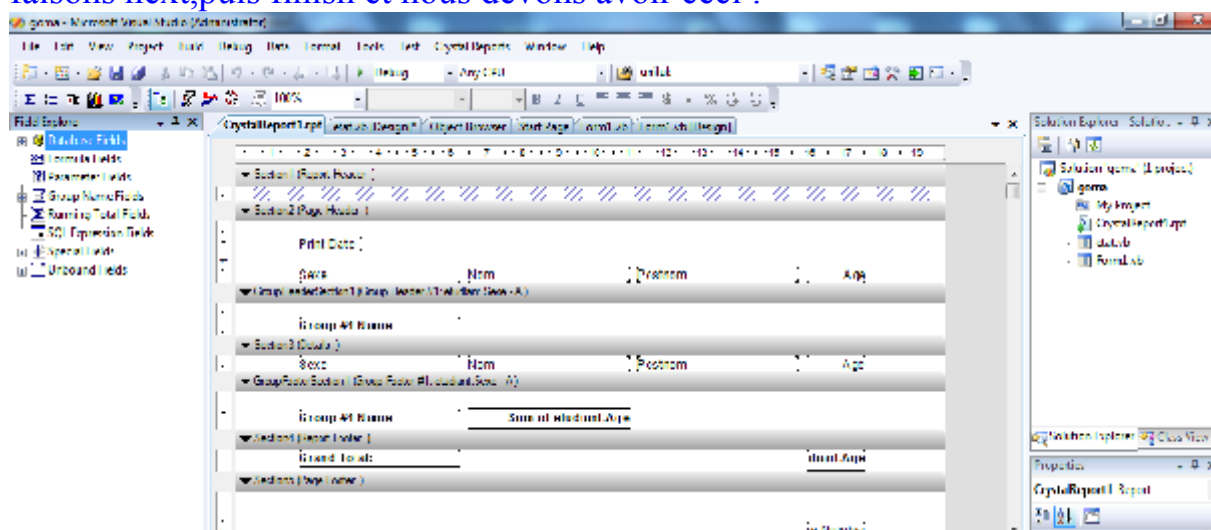
Selectionnons **Microsoft Jet 4.0 OLEDB Provider**, car c'est lui notre provider(cfr le driver pour la connexion)puis cliquons sur next. Selectionnons notre base de donnée qui se trouve dans «**C:/alimasi/masivi.mdb**», puis cliquons sans rien changé sur suivant et puis disons finish. Et voila notre nouvelle interface :



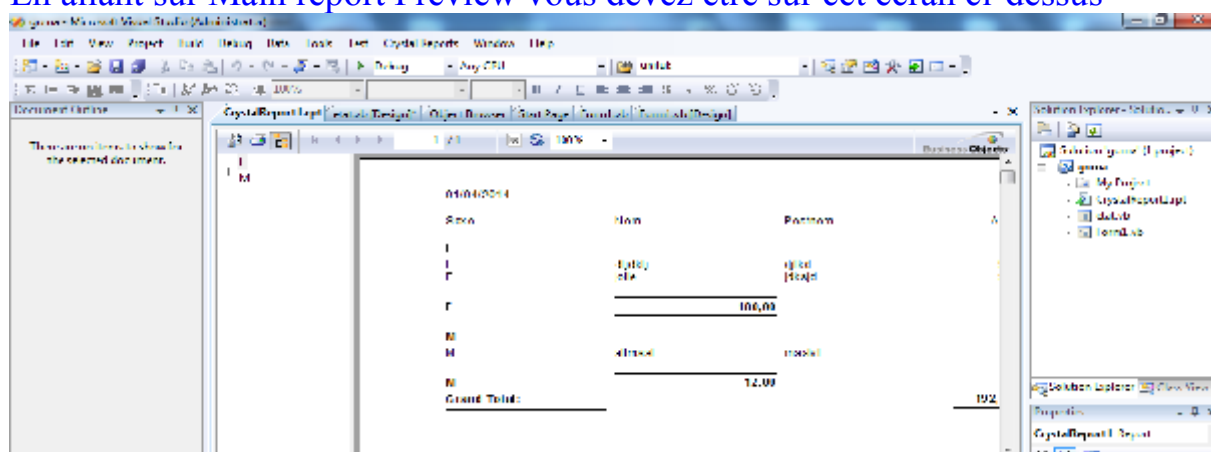
Selectionnons notre base de donnée etudiant, et avec le bouton selecteur, amenons le de l'autre coté pour avoir ceci enfin



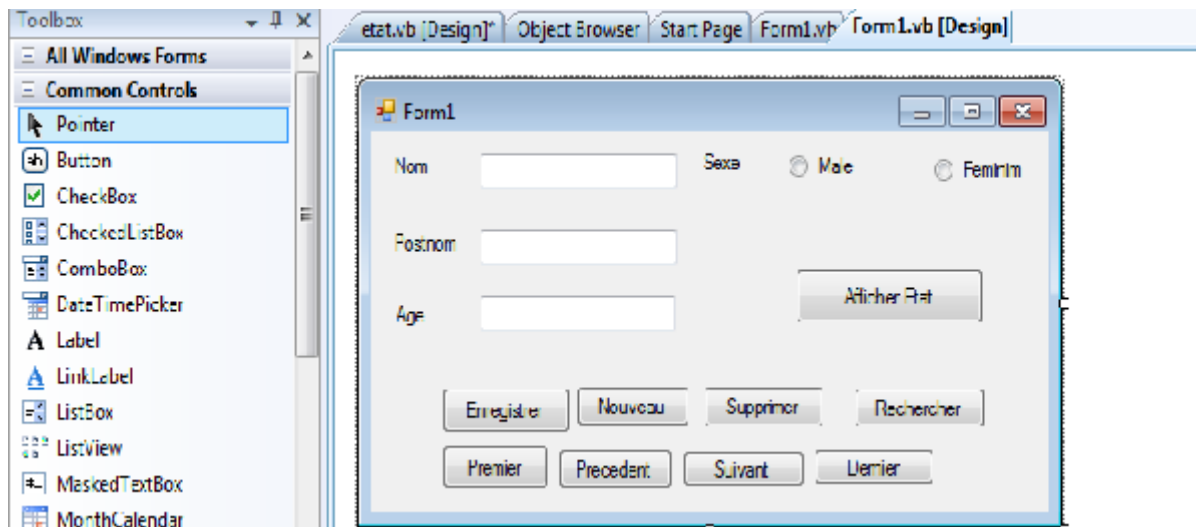
Puis faisons un next,selectionnons tout les champs, puis encore next, on peut grouper selon le sexe,puis next encore,ramenons dans le cote droit le champs age qui veut qu'on calcul sa somme comme il est du type numerique,puis faisons next,puis finish et nous devons avoir ceci :



En allant sur Main report Preview vous devez etre sur cet écran ci-dessus



Après ces étapes,ajoutons maintenant un bouton « afficher les données »sur notre formulaire, et voici ce qu'on doit avoir après :



Et puis cliquons sur le bouton « afficher état » pour écrire nos codes de la visualisation de notre report.

Try

```
'declaration de notre objet string qui contiendra notre instruction sql
Dim contenu As String
'initialisation de l'objet
contenu = ""
'instanciation de notre formulaire etat
Dim frm As New etat
'declaration de notre connexion
con.ConnectionString = ("Provider=Microsoft.Jet.OLEDB.4.0;Data
source=C:/alimasi/masivi.mdb")
'declaration d'un objet dataset
Dim donnees As New DataSet
donnees.Clear()
'ouverture de la connexion
con.Open()
'remplissage de notre variable avec l'instruction sql
'et c'est ici qu'on peut appliquer un where si on a pas envie de
selectionner
'tout les donnees
contenu = "SELECT * from etudiant"
'declaration d'un objet dataadaptateur
Dim adaptateur As OleDbDataAdapter
adaptateur = New OleDbDataAdapter(contenu, con)
```

```

adaptateur.Fill(donnees, "masivi")
'fermeture de la connexion
con.Close()
'instanciation de notre crystalreports
Dim objRpt As New CrystalReport1
objRpt.SetDataSource(donnees.Tables(0))
frm.CrystalReportViewer1.ReportSource = objRpt
frm.CrystalReportViewer1.RefreshReport()
frm.CrystalReportViewer1.Update()
'affichage de notre crystallreport
frm.Show()
Catch ex As Exception
    MsgBox("erreur")
End Try

```

Executons enfin notre formulaire et vous verrez que notre état sera visible.

Voici les codes au complet

```

Imports System.Data.OleDb
Imports System.Data
Public Class Form1
    Dim con As New OleDbConnection
    Dim forest As String
    Dim ds As New DataSet
    Dim da As OleDbDataAdapter
    'variable pour contenir le sexe Masculin et feminin
    Dim m As String
    Dim maxrows, x As Integer
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'declaration de la variable de la connexion
        con.ConnectionString = ("Provider=Microsoft.Jet.OLEDB.4.0;Data
source=C:/alimasi/masivi.mdb")
        'ouverture de la connexion
        con.Open()
        'selections des données
        forest = "select * from etudiant"
        'mettre les données dans le dataadaptateur
        da = New OleDb.OleDbDataAdapter(forest, con)
        'creation d'un lien entre la bd et le dataset
        da.Fill(ds, "masivi")
    End Sub
End Class

```

```

'fermeture de la connexion
con.Close()
'selection de toute nos lignes
maxrows = ds.Tables("masivi").Rows.Count
'initialisation de nos lignes
x = 0
'appel de notre procedure pour afficher les données
osee()
End Sub
Private Sub osee()
'fonction pour affectation données dans nos zones de textes
TextBox1.Text = ds.Tables("masivi").Rows(x).Item(0)
TextBox2.Text = ds.Tables("masivi").Rows(x).Item(1)
TextBox3.Text = ds.Tables("masivi").Rows(x).Item(3)
If ds.Tables("masivi").Rows(x).Item(2) = "M" Then
    RadioButton1.Checked = True
Else
    RadioButton2.Checked = True
End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles enregistrer.Click
' verification s'il n'ya pas des champs vide
Try
If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = ""
Then
    MsgBox("verifiez les champs vide")
Else
' ouverture de la connexion
con.Open()
'declarons un object command qui va modifier la base de donnee
Dim ajouter As New OleDbCommand
'code sql pour l'enregistrement des donnes
ajouter.CommandText = "insert into etudiant values (" &
TextBox1.Text & ", " & TextBox2.Text & ", " & m & ", " & TextBox3.Text &
")"

ajouter.Connection = con
ajouter.CommandType = CommandType.Text
ajouter.ExecuteNonQuery()
MsgBox("L'enregistrement a été ajouté avec succès")

```

```

        con.Close()
    End If
    Catch ex As Exception
        MsgBox("Erreur System")
    End Try
End Sub

```

```

Private Sub RadioButton1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged
    m = "M"
End Sub

```

```

Private Sub RadioButton2_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.CheckedChanged
    m = "F"
End Sub

```

```

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles nouveau.Click
    'initialisation des zones de textes
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    RadioButton1.Checked = False
    RadioButton2.Checked = False
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles supprimer.Click
    'ouverture de la connexion
    con.Open()
    'declaration de la commande
    Dim delete As New OleDbCommand
    'declaration d'une variable qui contiendra notre inputbox
    Dim k As String
    k = InputBox("Entrez le Numero de l'etudiant a supprimer")
    If k <> "" Then
        delete.CommandText = "delete from etudiant where Nom=" & k & ""
        delete.Connection = con
    End If
End Sub

```

```

delete.CommandType = CommandType.Text
delete.ExecuteNonQuery()
MsgBox("Un etudiant a ete supprimer")
Else
MsgBox("Rien a supprimer")
End If
con.Close()
End Sub

```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rechercher.Click
'ouverture de notre connexion
con.Open()
'declaration d'un string pour contenir notre inputbox et d'un objet command
Dim rechercher As New OleDbCommand
Dim p As String
p = InputBox("Entrez le nom de l'etudiant ")
rechercher.CommandText = "select * from etudiant where Nom=" & p &
,,,,
rechercher.Connection = con
'declaration d'un objet datareader qui nous permettra de lire les donnes de la
bd
Dim dbreader As OleDbDataReader = rechercher.ExecuteReader
If dbreader.Read = True Then
TextBox1.Text = dbreader.Item(0)
TextBox2.Text = dbreader.Item(1)
If dbreader.Item(2) = "M" Then
RadioButton1.Checked = True
Else
RadioButton2.Checked = True
End If
TextBox3.Text = dbreader.Item(3)
End If
con.Close()
End Sub

```

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles premier.Click
If x <> 0 Then
x = 0
osee()

```

```
Else
    MsgBox("Vous etes deja au premier enregistrement")
End If
End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles de.Click
    If x < maxrows - 1 Then
        x = x + 1
        'notre procedure
        osee()
    Else
        MsgBox("Vous êtes au dernier enregistrement")
    End If
End Sub
```

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles u.Click
    If x > 0 Then
        x = x - 1
        'notre procedure
        osee()
    Else
        MsgBox("Vous êtes au premier enregistrement")
    End If
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles dernier.Click
    If x <> maxrows - 1 Then
        x = maxrows - 1
        osee()
    Else
        MsgBox("Vous etes deja au dernier enregistrement")
    End If
End Sub
```

```
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles afficheretat.Click
Try
    'declaration de notre objet string qui contiendra notre instruction sql
```

```
Dim contenu As String
'initialisation de l'objet
contenu = ""
'instanciation de notre formulaire etat
Dim frm As New etat
'declaration de notre connexion
con.ConnectionString = ("Provider=Microsoft.Jet.OLEDB.4.0;Data
source=C:/alimasi/masivi.mdb")
'declaration d'un objet dataset
Dim donnees As New DataSet
donnees.Clear()
'ouverture de la connexion
con.Open()
'remplissage de notre variable avec l'instruction sql
'et c'est ici qu'on peut appliquer un where si on a pas envie de
selectionner
'tout les donnees
contenu = "SELECT * from etudiant"
'declaration d'un objet dataadaptateur
Dim adaptateur As OleDbDataAdapter
adaptateur = New OleDb.OleDbDataAdapter(contenu, con)
adaptateur.Fill(donnees, "masivi")
'fermeture de la connexion
con.Close()
'instanciation de notre crystalreports
Dim objRpt As New CrystalReport1
objRpt.SetDataSource(donnees.Tables(0))
frm.CrystalReportViewer1.ReportSource = objRpt
frm.CrystalReportViewer1.RefreshReport()
frm.CrystalReportViewer1.Update()
'affichage de notre crystallreport
frm.Show()
Catch ex As Exception
MsgBox("erreur")
End Try
End Sub
End Class
```