

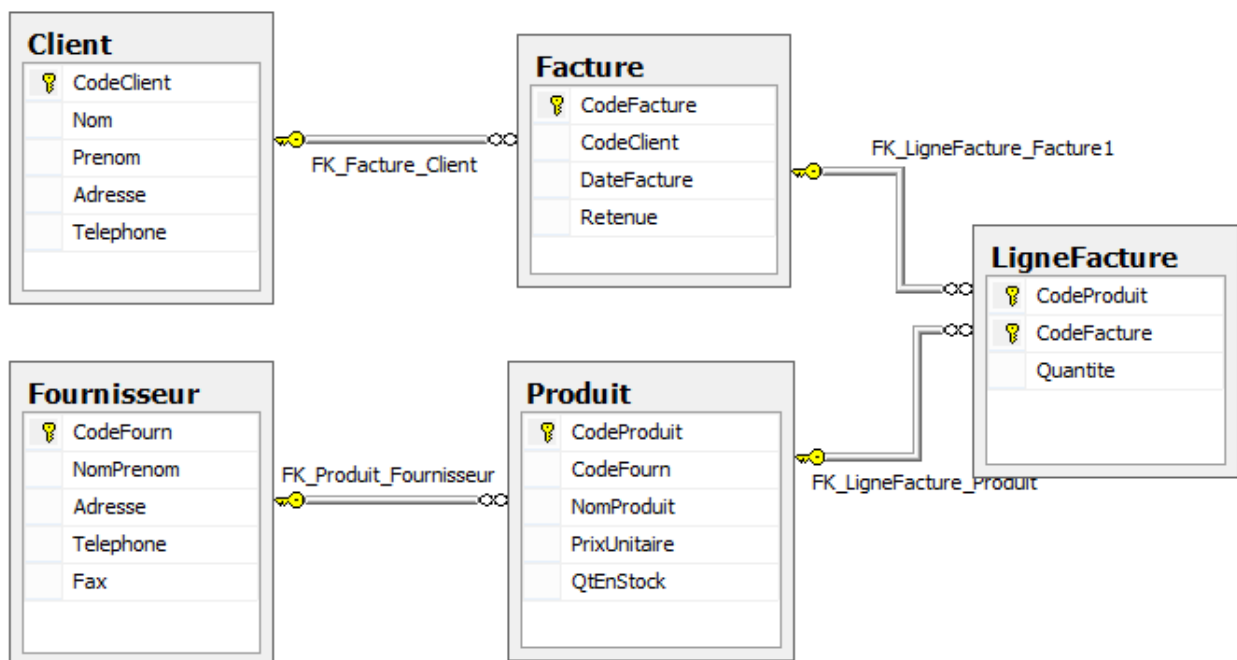
TP 6: ACCES AUX DONNEES

OBJECTIFS :

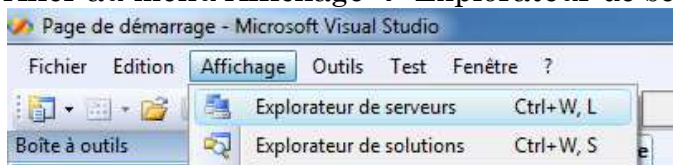
Apprendre la création d'une base de données sous SQL server et tester les manipulations de base sur la base de données.

I- Création de la base de données:

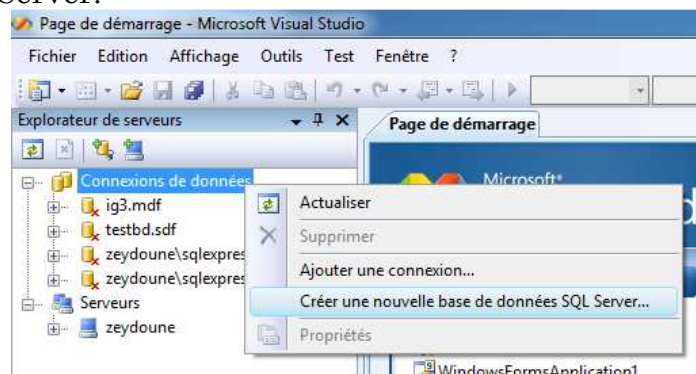
On veut créer une base de données sous SQLServer nommée **BDStock** contenant les tables suivantes:



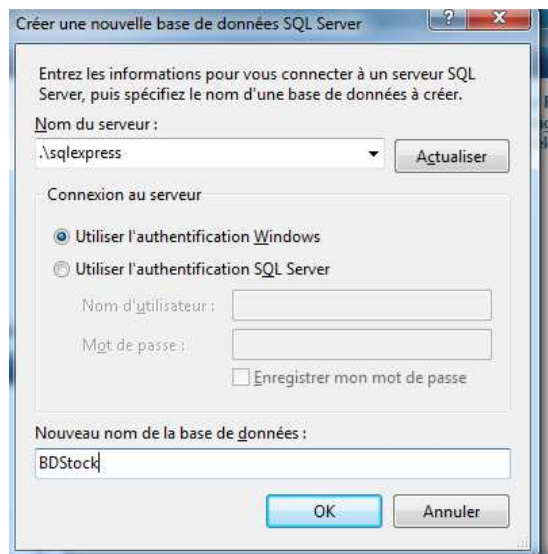
1. Démarrer Visual Studio 2008
2. Aller au menu Affichage → Explorateur de serveurs



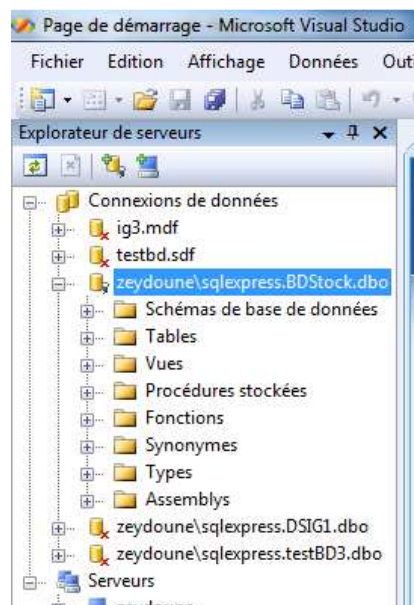
3. Dans la fenêtre Explorateur de serveurs, cliquer avec le bouton droite de la souris su Connexions de données et cliquer sur Créer une nouvelle base de données SQL Server.



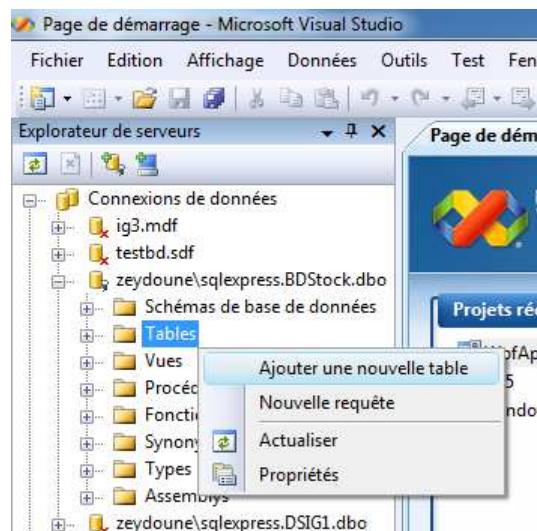
4. Dans la boîte de dialogue, le nom de serveur = .\sqlexpress et nom de la base de données = DBStock et cliquer sur ok.



5. La base est ajoutée dans la fenêtre Explorateur de serveurs.



6. Cliquez droit sur Tables et sélectionnez Ajouter une nouvelle table.



- Créer la table **Client**:

dbo.Client : Tabl...qlxpress.BDStock)			
	Nom de la colonne	Type de données	Null autorisé
🔑	CodeClient	int	<input type="checkbox"/>
	Nom	text	<input checked="" type="checkbox"/>
	Prenom	text	<input checked="" type="checkbox"/>
	Adresse	text	<input checked="" type="checkbox"/>
	Telephone	text	<input checked="" type="checkbox"/>

- Cliquez droite sur le champ *CodeClient* et sélectionner *Définir la clé primaire*

	Nom de la colonne	Type de données	Null autorisé
▶	CodeClient		
	Nom		
	Prenom		

- 🔑 Définir la clé primaire
- 🔗 Insérer une colonne
- 🗑 Supprimer une colonne

- Cliquer sur le bouton Enregistrer

Choisir un nom ? ✕

Entrez un nom pour la table :

Client

- Créer de la même manière la table **Facture**:

dbo.Facture : Tab...lexpress.BDStock)			
	Nom de la colonne	Type de données	Null autorisé
🔑	CodeFacture	bigint	<input type="checkbox"/>
	CodeClient	int	<input type="checkbox"/>
	DateFacture	datetime	<input checked="" type="checkbox"/>
	Retenue	float	<input checked="" type="checkbox"/>

- Créer la table **Fournisseur**:

dbo.Fournisseur : ...qlxpress.BDStock)			
	Nom de la colonne	Type de données	Null autorisé
🔑	CodeFourn	int	<input type="checkbox"/>
	NomPrenom	text	<input checked="" type="checkbox"/>
	Adresse	text	<input checked="" type="checkbox"/>
	Telephone	text	<input checked="" type="checkbox"/>
	Fax	text	<input checked="" type="checkbox"/>

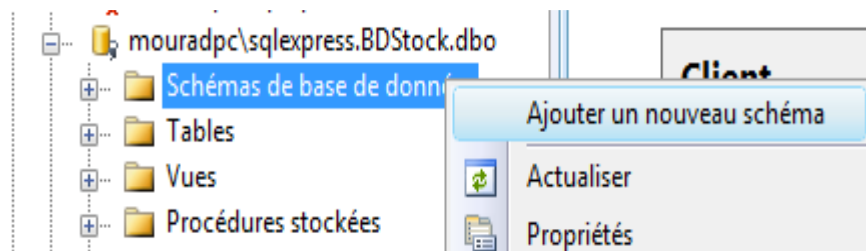
- Créer la table **Produit**:

dbo.Produit : Tabl...qlexpress.BDStock)			
	Nom de la colonne	Type de données	Null autorisé
🔑	CodeProduit	int	<input type="checkbox"/>
	CodeFourn	int	<input type="checkbox"/>
	NomProduit	text	<input checked="" type="checkbox"/>
	PrixUnitaire	text	<input checked="" type="checkbox"/>
	QtEnStock	int	<input checked="" type="checkbox"/>

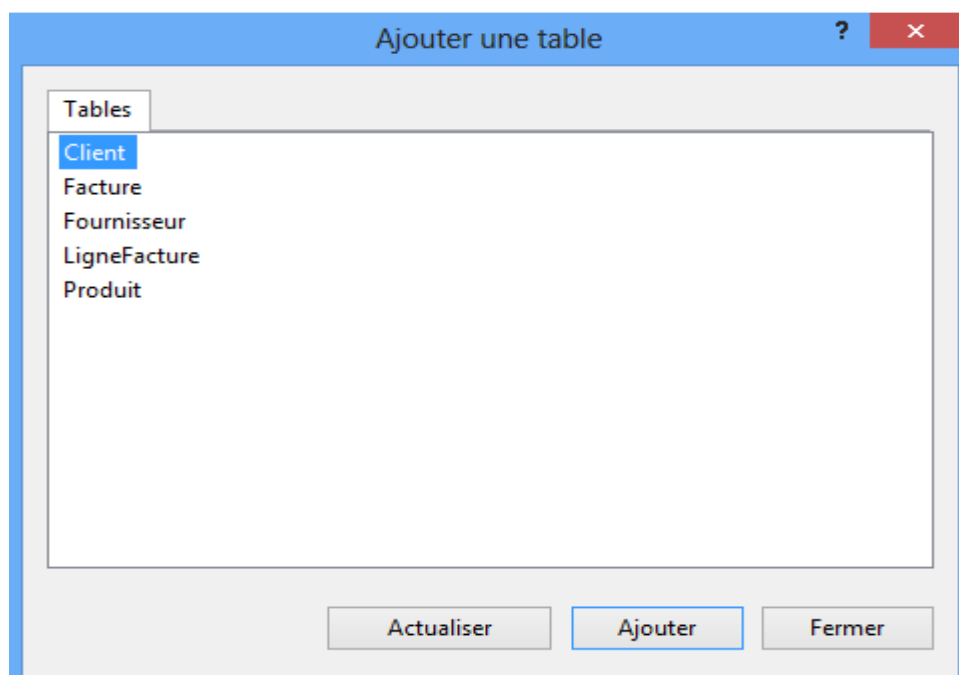
- Créer la table **LigneFacture**:

dbo.LigneFacture...lexpress.BDStock)			
	Nom de la colonne	Type de données	Null autorisé
🔑	CodeFacture	bigint	<input type="checkbox"/>
🔑	CodeProduit	int	<input type="checkbox"/>
	Quantite	int	<input checked="" type="checkbox"/>

7. Ajouter un nouveau schéma pour relier les différentes tables comme indiqué au début de l'énoncé.

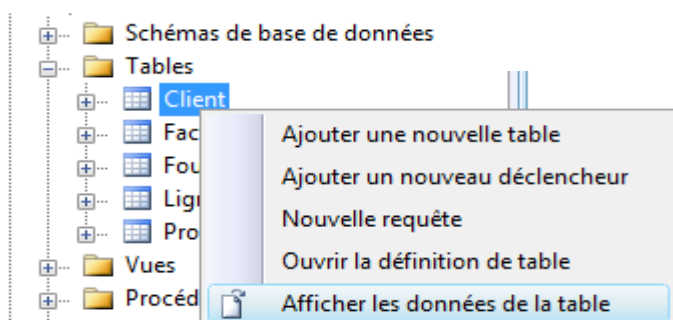


Ajouter les différentes tables, et avec le glissement de la souris ajouter les relations qui représentent les clés étrangères des tables.



II- Création de la base de données:

Charger les tables avec quelques données:



– La table **Client**:

	CodeClient	Nom	Prenom	Adresse	Telephone
	1	Suissi	Ali	Nabeul	72111111
	2	Ben Saleh	Sami	Tunis	71888888

– La table **Fournisseur**:

	CodeFourn	NomPrenom	Adresse	Telephone	Fax
	1	BenAhmed Maher	Tunis	71555555	NULL
	2	Khalidi Samir	Ariana	72666666	72666555

– La table **Produit**:

	CodeProduit	CodeFourn	NomProduit	PrixUnitaire	QtEnStock
	1	1	Clavier	13	30
	2	1	Souris	8	50
	3	2	Ecran17	180	10
	4	2	WebCam	20	25
	5	2	Flash 8Go	30	18

– La table **Facture**:

	CodeFacture	CodeClient	DateFacture	Retenue
	1	1	12/05/2012 ...	0
	2	2	11/11/2013 ...	10
	3	1	10/10/2014 ...	8

– La table **LigneFacture**:

	CodeFacture	CodeProduit	Quantite
	1	1	2
	1	2	4
	1	3	1
	2	1	1
	2	2	5
	2	4	6

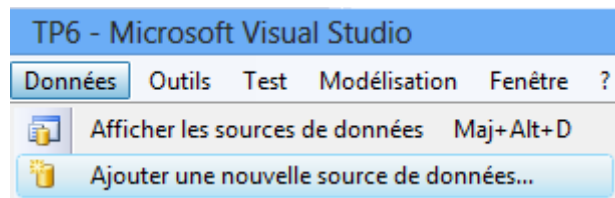
III- Création de l'interface d'accueil:

On veut réaliser une application qui permet de gérer le stock des produits ainsi que les différentes factures fournies aux clients, pour cela on vous demande de créer l'interface principale suivante:

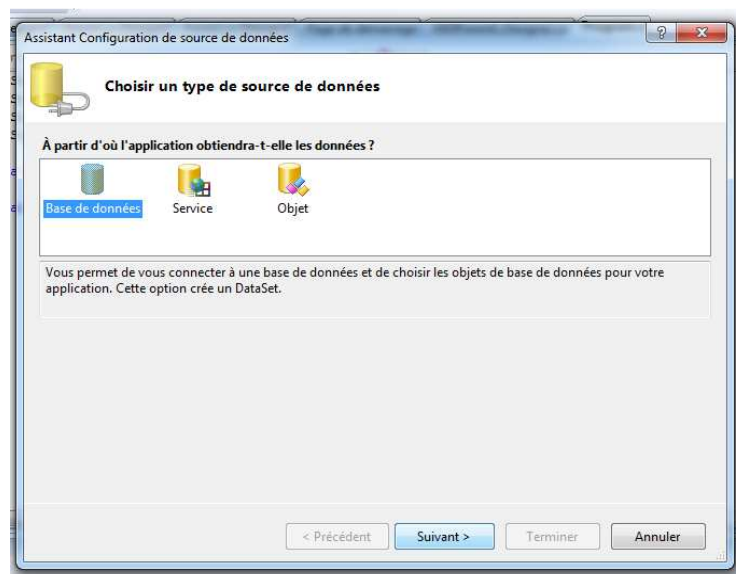


IV- Connexion avec la base de données:

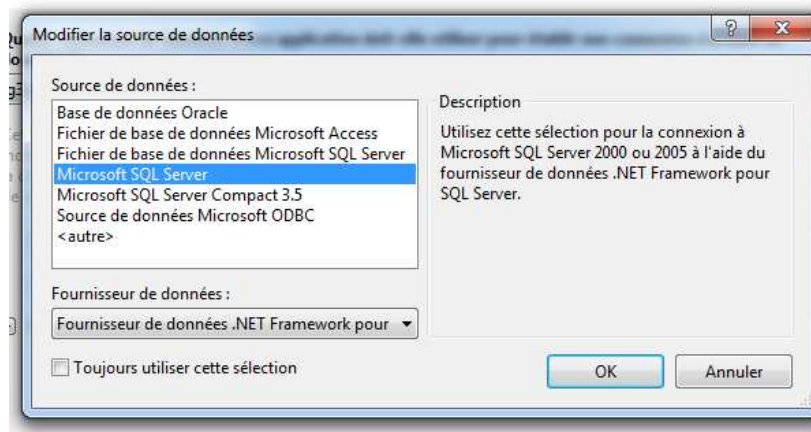
1. Aller au menu « Données » → « Ajouter une nouvelle source de données »



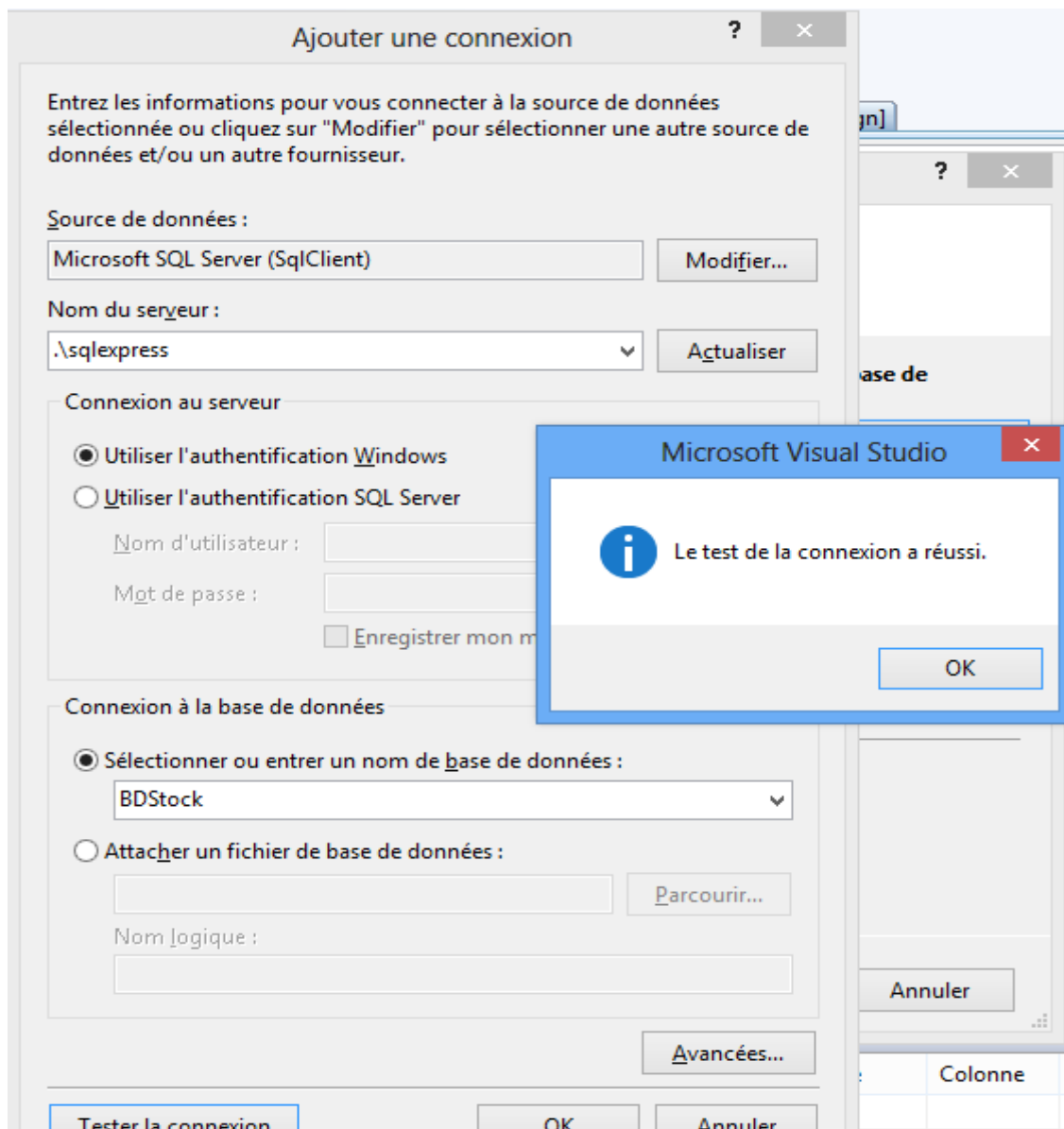
2. Sélectionner « Base de données » et cliquer sur « Suivant »



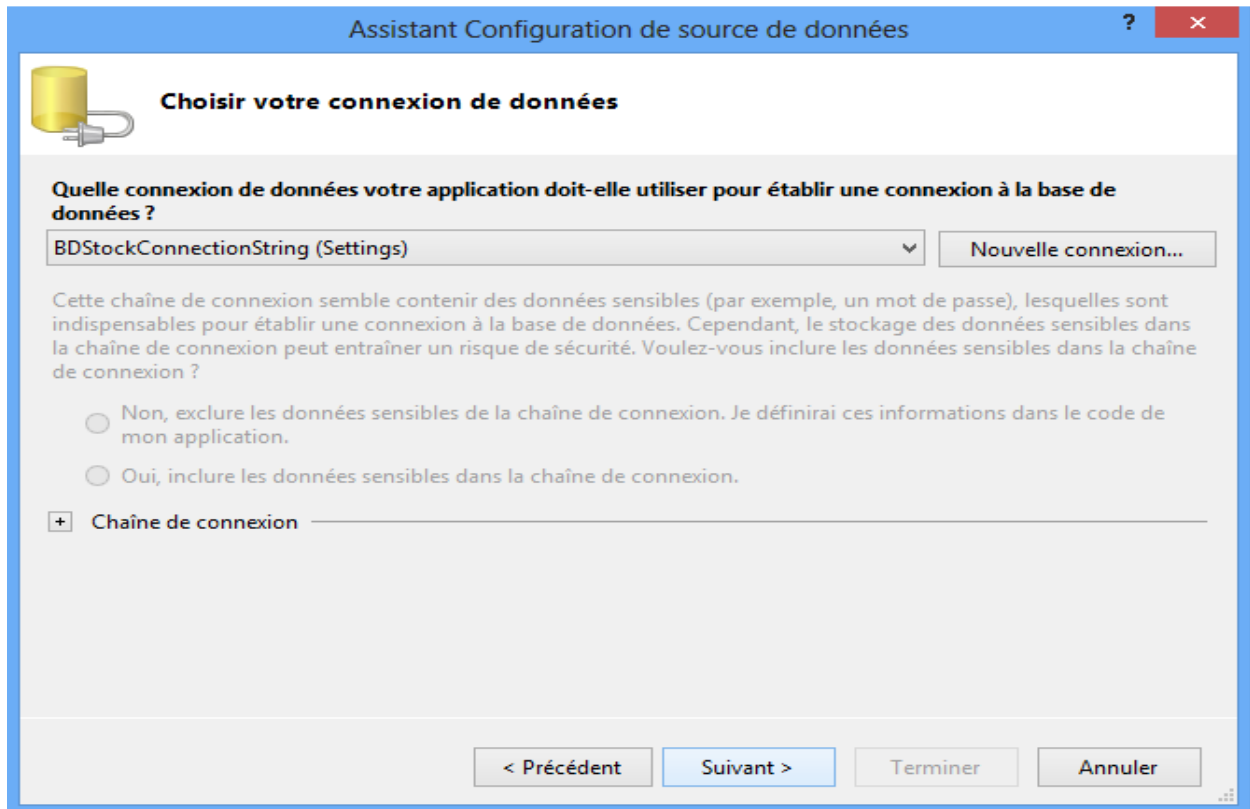
3. Cliquer sur « Nouvelle connexion »
4. Cliquer sur « Modifier »
5. Sélectionner « Microsoft SQL Server » et cliquer sur « OK »



6. Dans Nom du serveur, saisir « .\sqlexpress » et dans Sélectionner ou entrer un nom de base de données, choisir « BDStock » et cliquer sur « Tester la connexion ». Si la connexion est réussie, cliquer sur « ok »



7. Cliquer sur « Suivant »



Assistant Configuration de source de données

Choisir votre connexion de données

Quelle connexion de données votre application doit-elle utiliser pour établir une connexion à la base de données ?

BDStockConnectionString (Settings) Nouvelle connexion...

Cette chaîne de connexion semble contenir des données sensibles (par exemple, un mot de passe), lesquelles sont indispensables pour établir une connexion à la base de données. Cependant, le stockage des données sensibles dans la chaîne de connexion peut entraîner un risque de sécurité. Voulez-vous inclure les données sensibles dans la chaîne de connexion ?

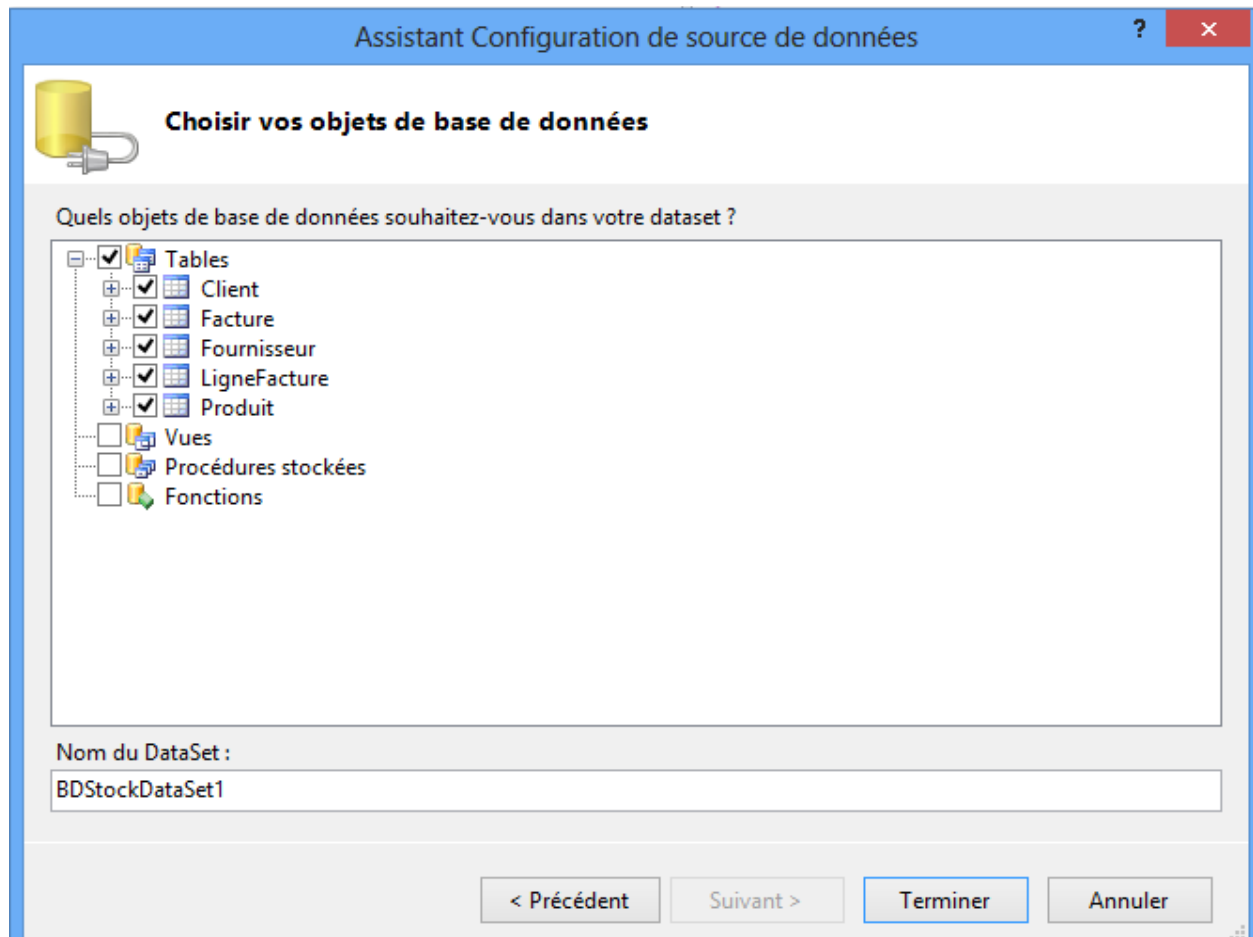
Non, exclure les données sensibles de la chaîne de connexion. Je définirai ces informations dans le code de mon application.

Oui, inclure les données sensibles dans la chaîne de connexion.

+ Chaîne de connexion

< Précédent Suivant > Terminer Annuler

8. Cocher « Tables » et cliquer sur « Terminer »



Assistant Configuration de source de données

Choisir vos objets de base de données

Quels objets de base de données souhaitez-vous dans votre dataset ?

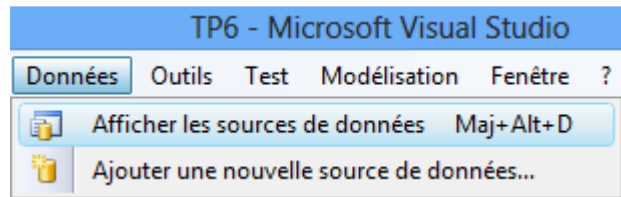
- Tables
 - Client
 - Facture
 - Fournisseur
 - LigneFacture
 - Produit
- Vues
- Procédures stockées
- Fonctions

Nom du DataSet :

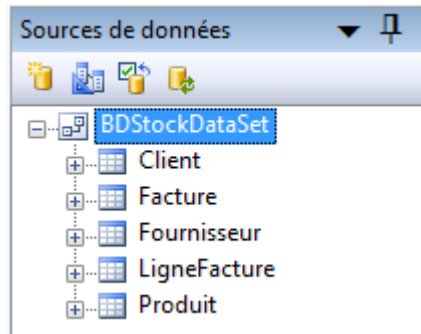
BDStockDataSet1

< Précédent Suivant > Terminer Annuler

9. Aller au menu « Données » → « Afficher les sources de données »

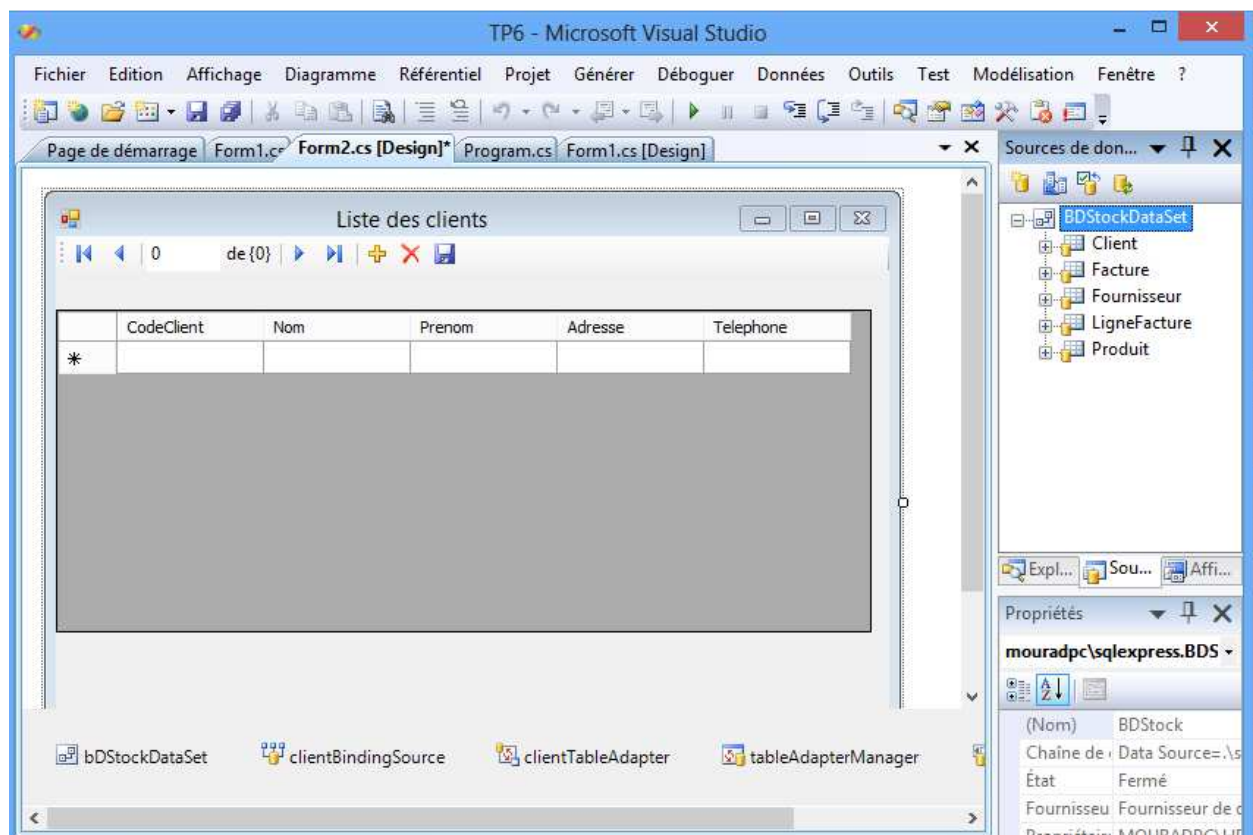


10. La fenêtre source de données s'affiche



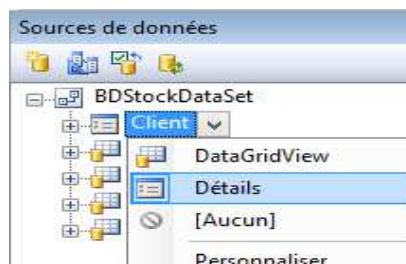
V- Gestion des Clients:

1. Créer une nouvelle interface « frmListeClients » et faire glisser la table « Client » à partir de la fenêtre « Sources de données » vers la forme. Le résultat sera le suivant:

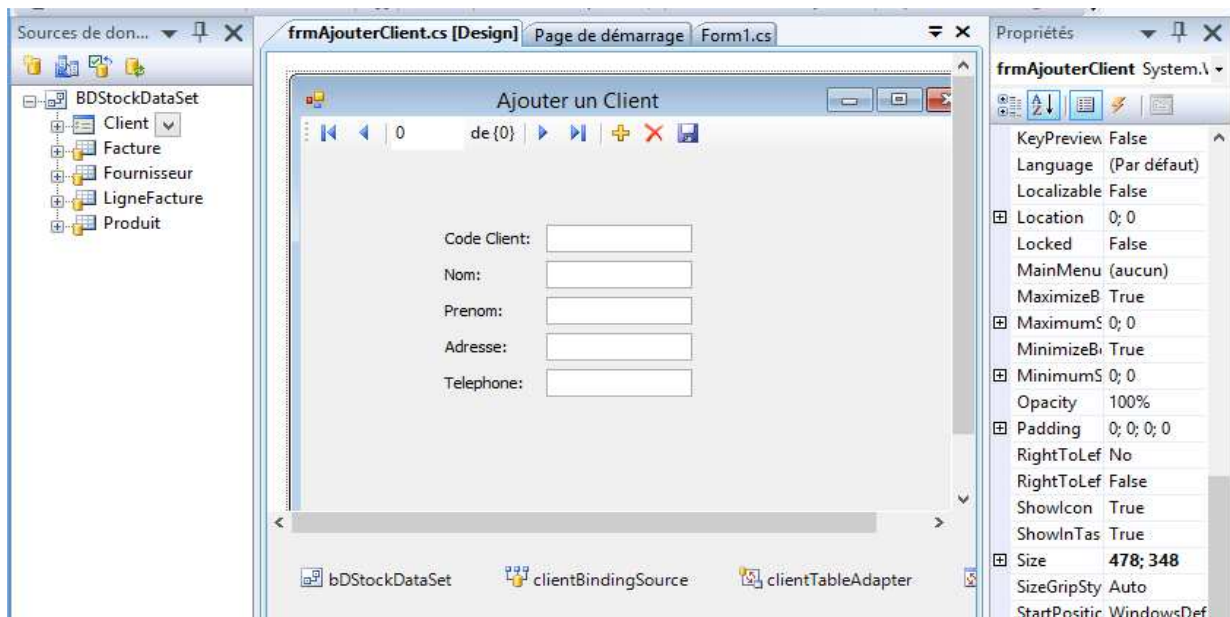


2. Sélectionner le dataGridView et aller à la fenêtre propriétés. Mettre la propriété Anchor = « Top, Bottom, Left, Right » et la propriété AutoSizeColoumsMode = « Fill »

3. Dans la fenêtre « Sources de données », cliquer sur le petit bouton à droite de la table « Client » et sélectionner dans la liste le mot « Détails »



4. Créer un nouveau formulaire « frmAjouterClient » et faire glisser la table « Client » à partir de la fenêtre « Sources de données » vers le formulaire:



V- Manipulation des fournisseur en mode déconnecté:

1. Créer un nouveau formulaire « frmAjouterFourn » , il faut mettre en place les contrôles à partir de la boîte à outils:



Indication sur le code:

- Bibliothèque à ajouter:

```
using System.Data.SqlClient;
```

- Déclaration globale:

```
SqlConnection cnx;
SqlCommand cmd;
SqlDataAdapter adap;
SqlCommandBuilder cb;
DataSet dset;
DataTable tab;
DataRow dtr;
int indice;
```

- Chargement de l'interface.:

```
cnx = new SqlConnection();
//connexion avec le serveurs de base de données
cnx.ConnectionString="Data Source=.\sqlexpress;Initial
Catalog=BDStock;Integrated Security=True";
//Ouverture de la connexion
cnx.Open();

//commande de selection des données
cmd=new SqlCommand();
cmd.CommandText="select * from Fournisseur";
cmd.Connection=cnx;

//chargement des données dans une table
adap=new SqlDataAdapter(cmd);
dset=new DataSet();
adap.Fill(dset,"Fournisseur");
tab=dset.Tables[0];

//chargement des données dans l'interface
indice=0;
txtCode.Text=(tab.Rows[0][0]).ToString();
txtNom.Text=(tab.Rows[0][1]).ToString();
txtAdr.Text=(tab.Rows[0][2]).ToString();
txtTel.Text=(tab.Rows[0][3]).ToString();
txtFax.Text=(tab.Rows[0][4]).ToString();
BTajout.Enabled = false;
```

- Le bouton nouveau:

```
txtCode.Text = "";
txtNom.Text = "";
txtAdr.Text = "";
txtTel.Text = "";
txtFax.Text = "";
BTajout.Enabled = true;
```

- Le bouton Premier:

```
indice = 0;
txtCode.Text = (tab.Rows[0][0]).ToString();
txtNom.Text = (tab.Rows[0][1]).ToString();
txtAdr.Text = (tab.Rows[0][2]).ToString();
txtTel.Text = (tab.Rows[0][3]).ToString();
txtFax.Text = (tab.Rows[0][4]).ToString();
```

- Le bouton Précédent:

```
if (indice > 0)
{
    indice--;
    txtCode.Text = (tab.Rows[indice][0]).ToString();
    txtNom.Text = (tab.Rows[indice][1]).ToString();
    txtAdr.Text = (tab.Rows[indice][2]).ToString();
    txtTel.Text = (tab.Rows[indice][3]).ToString();
    txtFax.Text = (tab.Rows[indice][4]).ToString();
}
```

- Le bouton Suivant:

```
if (indice < tab.Rows.Count - 1)
{
    indice++;
    txtCode.Text = (tab.Rows[indice][0]).ToString();
    txtNom.Text = (tab.Rows[indice][1]).ToString();
    txtAdr.Text = (tab.Rows[indice][2]).ToString();
    txtTel.Text = (tab.Rows[indice][3]).ToString();
    txtFax.Text = (tab.Rows[indice][4]).ToString();
}
```

- Le bouton Dernier:

```
indice = tab.Rows.Count - 1;
txtCode.Text = (tab.Rows[indice][0]).ToString();
txtNom.Text = (tab.Rows[indice][1]).ToString();
txtAdr.Text = (tab.Rows[indice][2]).ToString();
txtTel.Text = (tab.Rows[indice][3]).ToString();
txtFax.Text = (tab.Rows[indice][4]).ToString();
```

- Le bouton Ajouter:

```
dtr = tab.NewRow();
dtr[0] = txtCode.Text;
dtr[1] = txtNom.Text;
dtr[2] = txtAdr.Text;
dtr[3] = txtTel.Text;
dtr[4] = txtFax.Text;
dset.Tables[0].Rows.Add(dtr);
tab = dset.Tables[0];
indice = tab.Rows.Count - 1;
MessageBox.Show("Fournisseur ajouté");
BTajout.Enabled = false;
```

- Le bouton Supprimer:

```
try
{
    if (MessageBox.Show("Voulez vous vraiment supprimer ce
fournisseur?", "Confirmer suppression", MessageBoxButtons.YesNo)
== DialogResult.Yes)
    {
        dset.Tables[0].Rows[indice].Delete();
        dset.AcceptChanges();
        indice--;
        tab = dset.Tables[0];
    }
}
```

```

        if (indice<0)
            indice=0;
    if (tab.Rows.Count != 0)
    {
        txtCode.Text = (tab.Rows[indice][0]).ToString();
        txtNom.Text = (tab.Rows[indice][1]).ToString();
        txtAdr.Text = (tab.Rows[indice][2]).ToString();
        txtTel.Text = (tab.Rows[indice][3]).ToString();
        txtFax.Text = (tab.Rows[indice][4]).ToString();
    }
    else
    {
        txtCode.Text = "";
        txtNom.Text = "";
        txtAdr.Text = "";
        txtTel.Text = "";
        txtFax.Text = "";
    }
    MessageBox.Show("Fournisseur Supprimé");
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

- Le bouton Modifier:

```

try
{
    dset.Tables[0].Rows[indice][0] = txtCode.Text;
    dset.Tables[0].Rows[indice][1] = txtNom.Text;
    dset.Tables[0].Rows[indice][2] = txtAdr.Text;
    dset.Tables[0].Rows[indice][3] = txtTel.Text;
    dset.Tables[0].Rows[indice][4] = txtFax.Text;
    tab = dset.Tables[0];
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

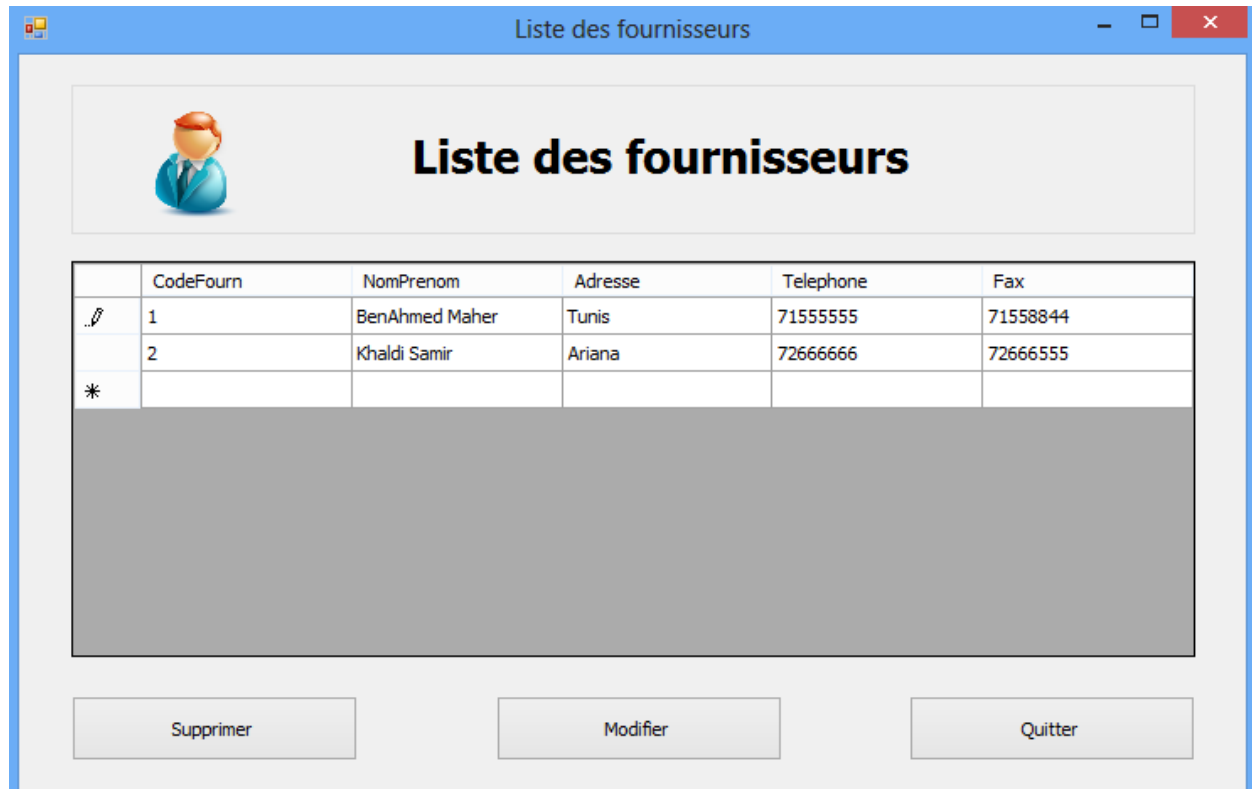
- Le bouton Enregistrer:

```

try
{
    cb = new SqlCommandBuilder(adap);
    adap.Update(dset, "Fournisseur");
    dset.Clear();
    adap.Fill(dset, "Fournisseur");
    tab = dset.Tables[0];
    MessageBox.Show("Données enregistrées dans la base");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

2. Créer l'interface suivante permettant de gérer la liste des fournisseurs.



Indication sur le code:

- Bibliothèque à ajouter:

```
using System.Data.SqlClient;
```

- Déclaration globale:

```
SqlConnection cnx;
SqlCommand cmd;
SqlDataAdapter adap;
SqlCommandBuilder cb;
DataSet dset;
```

- Chargement de l'interface:

```
cnx = new SqlConnection();
cnx.ConnectionString="Data Source=.\sqlexpress;Initial
Catalog=BDStock;Integrated Security=True";
cnx.Open();
cmd=new SqlCommand();
cmd.CommandText="select * from Fournisseur";
cmd.Connection=cnx;
adap=new SqlDataAdapter(cmd);
dset=new DataSet();
adap.Fill(dset,"Fournisseur");
dataGridView1.DataSource = dset.Tables[0];
```

- Le bouton supprimer:

```
try
{
    if (MessageBox.Show("Voulez vous vraiment supprimer ce
fournisseur?", "Confirmer suppression", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
    {
```

```

        int i = dataGridView1.CurrentRow.Index;
        dset.Tables[0].Rows[i].Delete();
        cb = new SqlCommandBuilder(adap);
        adap.Update(dset, "Fournisseur");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

- Le bouton Modifier:

```

try
{
    if (MessageBox.Show("Voulez vous enregistrer les
modifications?", "Confirmer modification", MessageBoxButtons.YesNo)
== DialogResult.Yes)
    {
        cb = new SqlCommandBuilder(adap);
        adap.Update(dset, "Fournisseur");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

VI- Manipulation des Produit en mode connecté:

1. Créer le nouveau formulaire « frmAjouterProd » suivant:

The screenshot shows a Windows application window titled "Ajouter un produit". The window has a light gray background and a blue title bar. At the top left, there is a green shopping basket icon. To its right, the text "Ajout d'un Produit" is displayed in a large, bold, black font. Below this, there are five input fields arranged vertically, each with a label to its left: "Code:", "Nom:", "Prix unitaire:", "Fournisseur:", and "Quantité:". The "Fournisseur:" field is a dropdown menu with "BenAhmed Maher" selected. At the bottom of the window, there are two buttons: "Ajouter" and "Quitter", both with a light gray background and black text.

Indication sur le code:▪ Chargement de l'interface:

```

cnx = new SqlConnection();
cnx.ConnectionString = "Data Source=.\sqlservr;Initial
Catalog=BDStock;Integrated Security=True";
cnx.Open();
cmd = new SqlCommand();
cmd.CommandText = "select * from Fournisseur";
cmd.Connection = cnx;
adap = new SqlDataAdapter(cmd);
dset = new DataSet();
adap.Fill(dset, "Fournisseur");
cmbFourn.DataSource = dset.Tables[0];
cmbFourn.DisplayMember = "NomPrenom";
cmbFourn.ValueMember = "CodeFourn";

```

▪ Le bouton Ajouter:

```

if (MessageBox.Show("Voulez vous vraiment ajouter ce produit ?",
"Ajout d'un Produit", MessageBoxButtons.YesNo) == DialogResult.Yes)
{
    try
    {
        cmd.CommandText = "insert into Produit values (" +
txtCode.Text + "," + cmbFourn.SelectedValue.ToString() + "," +
txtNom.Text + "," + txtPrix.Text + "," + txtQt.Text + ")";
        cmd.ExecuteNonQuery();
        MessageBox.Show("Insertion effectuée avec succes","Insertion");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

2. Créer le nouveau formulaire « frmListeProd » suivant:

CodeProduit	NomProduit	Fournisseur	QtUnitaire	QtEnStock
1	Clavier	BenAhmed Maher	25	30
2	Souris	BenAhmed Maher	8	50
3	Ecran17	Khalidi Samir	180	10
4	WebCam	Khalidi Samir	20	25
5	Flash 8Go	Khalidi Samir	30	18
66	66	Khalidi Samir	66	666
*				

Indication sur le code:

- Chargement de l'interface:

```

cnx = new SqlConnection();
cnx.ConnectionString = "Data Source=.\sqlexpress;Initial
Catalog=BDStock;Integrated Security=True";
cnx.Open();
cmd = new SqlCommand();
cmd.CommandText = "select * from Fournisseur";
cmd.Connection = cnx;
adap = new SqlDataAdapter(cmd);
dset = new DataSet();
adap.Fill(dset, "Fournisseur");
DataRow dr;
dr= dset.Tables[0].NewRow();
dr["NomPrenom"] = "Toute la liste";
dset.Tables[0].Rows.Add(dr);

cmbFourn.DataSource = dset.Tables[0];
cmbFourn.DisplayMember = "NomPrenom";
cmbFourn.ValueMember = "CodeFourn";

cmd = new SqlCommand();
cmd.CommandText = "select CodeProduit,NomProduit,NomPrenom as
'Fournisseur', PrixUnitaire, QtEnStock from Produit P, Fournisseur F
where P.CodeFourn=F.CodeFourn";
cmd.Connection = cnx;
adap = new SqlDataAdapter(cmd);
dset2 = new DataSet();
adap.Fill(dset2, "Produit");
dataGridView1.DataSource = dset2.Tables[0];
cmbFourn.SelectedIndex = cmbFourn.Items.Count - 1;

```

- Le comboBox des fournisseurs :

```

private void cmbFourn_SelectionChangeCommitted(...)
{
    cmd = new SqlCommand();
    if(cmbFourn.SelectedIndex==cmbFourn.Items.Count - 1)
        cmd.CommandText = "select
CodeProduit,NomProduit,NomPrenom as 'Fournisseur', PrixUnitaire,
QtEnStock from Produit P, Fournisseur F where
P.CodeFourn=F.CodeFourn";
    else
        cmd.CommandText = "select
CodeProduit,NomProduit,NomPrenom as 'Fournisseur', PrixUnitaire,
QtEnStock from Produit P, Fournisseur F where
P.CodeFourn=F.CodeFourn and P.CodeFourn=" + cmbFourn.SelectedValue;
    cmd.Connection = cnx;
    adap = new SqlDataAdapter(cmd);
    dset2 = new DataSet();
    adap.Fill(dset2, "Produit");
    dataGridView1.DataSource = dset2.Tables[0];
}

```

- Le bouton Supprimer :

```

    if (MessageBox.Show("Voulez vous vraiment supprimer ce produit
?", "Supprimer un Produit", MessageBoxButtons.YesNo) ==
DialogResult.Yes) {
    try
    {
        cmd.CommandText = "delete from Produit where CodeProduit=" +
dataGridView1.CurrentRow.Cells[0].Value.ToString();
        cmd.ExecuteNonQuery();
        MessageBox.Show("suppression effectuée avec succes", "Suppression");
        cmd = new SqlCommand();
        cmd.CommandText = "select CodeProduit,NomProduit,NomPrenom as
'Fournisseur', PrixUnitaire, QtEnStock from Produit P, Fournisseur F
where P.CodeFourn=F.CodeFourn";
        cmd.Connection = cnx;
        adap = new SqlDataAdapter(cmd);
        dset2 = new DataSet();
        adap.Fill(dset2, "Produit");
        dataGridView1.DataSource = dset2.Tables[0];
        cmbFourn.SelectedIndex = cmbFourn.Items.Count - 1;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

3. Créer le nouveau formulaire « frmFindProd » suivant:

	CodeProduit	NomProduit	Fournisseur	PrixUnitaire	QtEnStock
	5	Flash 8Go	Khaldi Samir	30	18
	6	Flash 4Go	BenAhmed Maher	10	40

```

if (rdbnom.Checked)
{
    cmd = new SqlCommand();
    cmd.CommandText = "select CodeProduit,NomProduit,NomPrenom as
'Fournisseur', PrixUnitaire, QtEnStock from Produit P, Fournisseur F
where P.CodeFourn=F.CodeFourn and NomProduit
like('"+textBox1.Text+"%')";
    .....
}

```