

Laboratoire 2

ADO.NET : le mode déconnecté en utilisant les assistants de Visual Studio

1. Objectifs d'apprentissage

- Se connecter à une source de données à l'aide de l'Assistant
- Visualiser les données d'une base de données sur le serveur
- Créer des interfaces simples de base de données à l'aide de l'Assistant
- Créer des groupes de données (des *DataTable*) dans un *DataSet*
- Travailler avec des *DataTable* et des *TableAdapter*
- Interroger et mettre à jour une base de données en mode déconnecté en utilisant les assistants

2. Créer une source de données

- Créez un nouveau projet nommé **Labo2** en utilisant le modèle *Application Windows Desktop* (en *Visual C#*)/Application Windows Forms
- Pour accéder à une base de données à partir d'un projet *.NET*, il faut d'abord créer une source de données comme suit :
 - À partir de la barre de menus verticale, cliquez sur *Sources de données* puis sur *Ajouter une nouvelle source de données*. L'assistant *Configuration de source de données* apparaît.
 - Choisissez *Base de données*, puis cliquez sur *Suivant*.
 - Sur la page *Choisir un modèle de base de données*, sélectionnez *DataSet* puis cliquez sur *Suivant*.
 - Sur la page *Choisir votre connexion de données*, sélectionnez *Nouvelle connexion...* la boîte de dialogue *Choisir la source de données* apparaît. Sélectionnez *Microsoft SQL Serveur* comme source de données puis cliquez sur *Continuer...* La boîte de dialogue *Ajouter une connexion* apparaît.
 - Tapez le nom du serveur **424-sql2017,5433** et le type d'authentification (sélectionnez l'authentification *SQL Serveur*). Tapez votre nom d'utilisateur et votre mot de passe. Sélectionnez *Enregistrer*

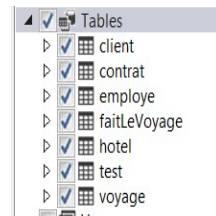
mon mot de passe.

- Cliquez sur *Tester la connexion.*
- Sélectionnez la base de données **BDVoyagesVotreNom** puis cliquez sur *OK.*
- Sur la page *Choisir votre connexion*, sélectionnez *Oui, inclure les données sensibles dans la chaîne de connexion.* Pour visualiser votre chaîne de connexion, cochez la case *Afficher la chaîne de connexion.....*
- Cliquez sur *Suivant.*
- Sur la page *Enregistrer la chaîne de connexion dans le fichier de configuration*, assurez-vous que la case *Oui, enregistrez la connexion en tant que :* est cochée.

Remarque 1 : **BDVoyagesVotreNomConnectionString** est le nom (par défaut) de la variable qui contient votre chaîne de connexion. Vous pouvez utiliser cette variable à tout moment dans votre application

Remarque 2 : Un projet C# sauvegarde la chaîne de connexion dans le fichier de configuration de l'application **app.config**. Ce fichier est créé automatiquement quand on crée une source de données.

- Cliquez sur *Suivant.*
- La page *Choisir vos objets de la base de données* s'affiche. Cochez la case *Tables* puis développez le nœud *Tables* pour vérifier que toutes les tables de votre base de données sont bien cochées.



Remarque 3 : Cette page vous permet de choisir les objets de la base de données (tables, vues, procédures stockées et fonctions) qui seront accessibles dans votre projet. Ces objets seront regroupés dans une structure de données fortement typée (le *DataSet*). Dans votre cas, le *DataSet* devrait porter le nom de **BDVoyagesVotreNomDataSet**. Ne modifiez pas ce nom.

Pour chaque table, l'assistant crée également un *TableAdapter* pour faire le lien entre la table de la base de données et celle du *DataSet* (un *DataTable*)

Le *DataSet* est stockée dans un fichier **.xsd** (XML schema définition).

- Cliquez sur *Terminer.*

3. Visualiser les données

Lorsque la source de données est créée, plusieurs outils peuvent être utilisés pour visualiser la structure et les données présentes dans la base de données.

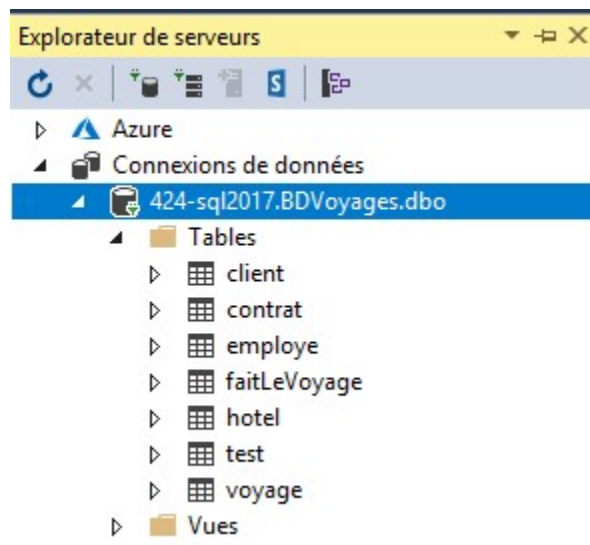
La chaîne de connexion

Pour visualiser votre chaîne de connexion, double-cliquez sur le fichier *app.config* dans l'*Explorateur de solutions*.

Observez que **BDVoyagesVotreNomConnectionString** est le nom (par défaut) de la variable qui contient votre chaîne de connexion.

L'Explorateur de serveurs

Pour visualiser la base de données réelle : à partir de la barre de menus verticale, cliquez sur *Explorateur de serveurs*.



À partir de l'*Explorateur de serveurs*, vous pouvez effectuer diverses opérations sur les objets de la base de données réelle. Par exemple : Cliquez-droit sur la table **client** puis sélectionnez *Afficher les données de la table*.

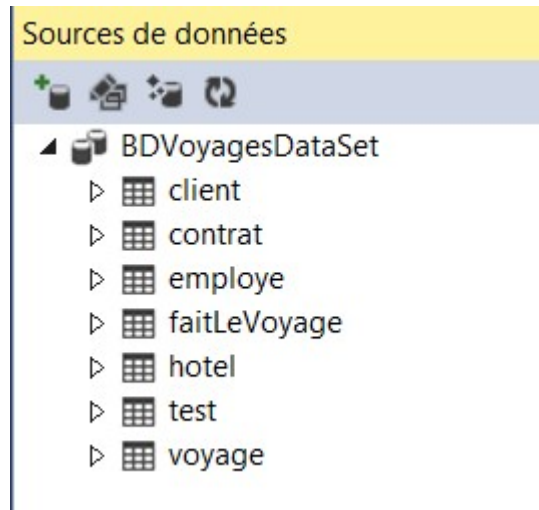
Vous pouvez aussi modifier, ajouter et supprimer des données à partir de l'*Explorateur de serveurs*.

Finalement, il est également possible d'ajouter des tables et d'exécuter des requêtes directement sur le serveur.

Dans ce cours, nous utiliserons l'*Explorateur de serveurs* pour vérifier la base de données (sans être obligé de passer par *SQL Server Management Studio*).

La fenêtre *Sources de données*

Pour visualiser cette fenêtre, cliquez sur *Sources de données*.



Remarque très importante

Il ne faut pas confondre le *DataSet* et votre base de données réelle.

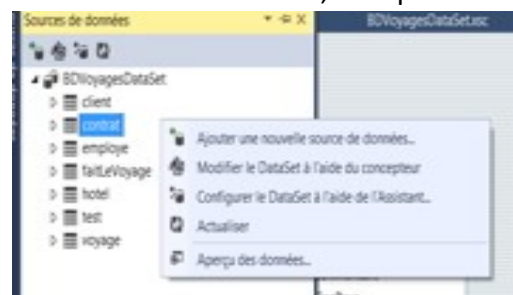
Le *DataSet* est situé en mémoire tandis que votre base de données réelle se trouve sur le serveur. De plus, le *DataSet* n'a pas exactement la même structure que votre base de données réelle. Il est possible que le *DataSet* ne contienne qu'une partie seulement de votre base de données réelle.

Toute modification sur le *DataSet* n'a aucun effet sur votre base de données réelle et vice et versa.

Il est fortement suggéré de ne pas modifier la structure de votre base de données réelle lorsque vous travaillez à l'aide d'un *DataSet*. Cela assure la cohérence entre les deux structures.

Si vous modifiez la structure de votre base de données réelle, vous allez devoir recréer votre *DataSet* au complet.

Si vous cliquez-droit n'importe où dans la fenêtre *Sources de données*, les opérations suivantes sont possibles.

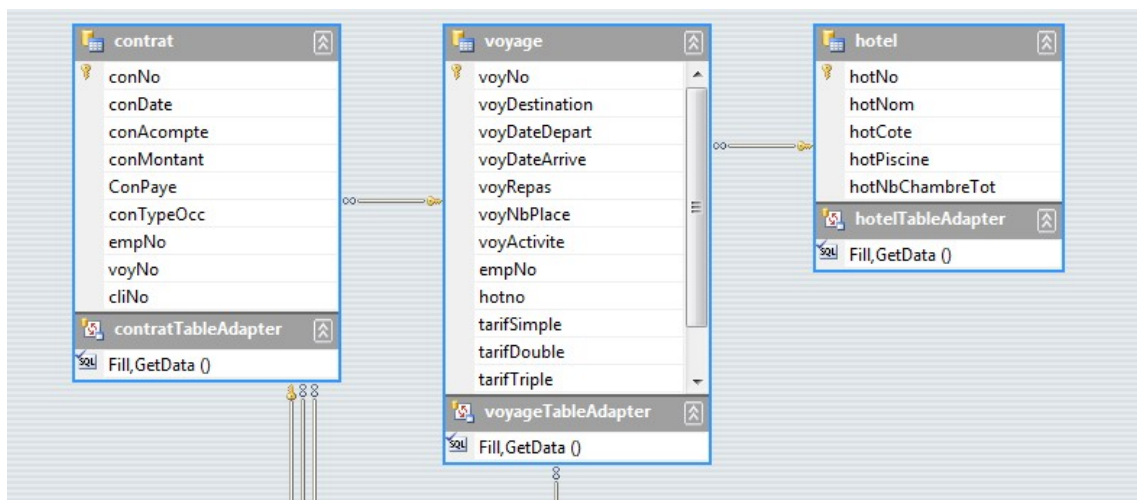


Remarque : On peut visualiser et modifier des données à partir de *l'Explorateur de serveurs*. Par contre, on ne peut que visualiser des données à partir de la fenêtre *Sources de données*.

Nous utiliserons la fenêtre *Sources de données* pour visualiser les données du *DataSet* ainsi que le *Concepteur de groupes de données* (de *DataTables*) pour créer des interfaces qui contiennent des contrôles liés aux données du *DataSet*.

La structure de la base de données dans le *DataSet*

Pour visualiser la structure de la base de données dans le *DataSet* ainsi que ses propriétés, double-cliquez sur le fichier *BDVoyagesVotreNomDataSet.xsd* dans *l'Explorateur de solutions*.



Observez, pour chacune des *DataTables* du *DataSet*, qu'il y a un *TableAdapter* qui lui est associée. C'est ce *TableAdapter* qui remplit les données de la *DataTable* correspondante à partir des données de la table réelle. C'est également grâce à ce *TableAdapter* que les données de la table réelle sont mises à jour lorsque celles de la *DataTable* correspondante sont modifiées dans le *DataSet*.

Vous pouvez visualiser les propriétés de chaque élément de votre *DataSet*.

Cliquez sur **contrat** pour voir les propriétés de la *DataTable* **contrat**.

Propriétés	
contrat DataTable	
CaseSensitive	False
Locale	Français (Canada)
Name	contrat

Cliquez sur le champ **conNo** pour voir les propriétés de la *DataColumn* **conNo** de la *DataTable* **contrat**.

Propriétés

conNo DataColumn

AutoIncrementSeed	0
AutoIncrementStep	1
Caption	conNo
DataType	System.Decimal
DateTimeMode	UnspecifiedLocal
DefaultValue	<DBNull>

Par exemple, double-cliquez sur le lien qu'il y a entre la *DataTable* **contrat** et la *DataTable* **voyage** pour voir les propriétés de la relation qu'il y a entre ces 2 *DataTables*.

Relation

Nom :

Spécifiez les clés qui relient les tables dans votre dataset.

Table parente : Table enfant :

Colonnes clés	Colonnes clés étrangères
voyNo	voyNo

Choisissez ce que vous voulez créer

☐ Relation et contrainte de clé étrangère
☐ Contrainte de clé étrangère uniquement
☒ Relation uniquement

Règle de mise à jour :

Règle de suppression :

Règle d'acceptation/de rejet :

☐ Relation imbriquée

OK Annuler

Cliquez sur le *TableAdapter* **contratTableAdapter** pour voir les propriétés de ce *TableAdapter*.

Propriétés

contratTableAdapter TableAdapter

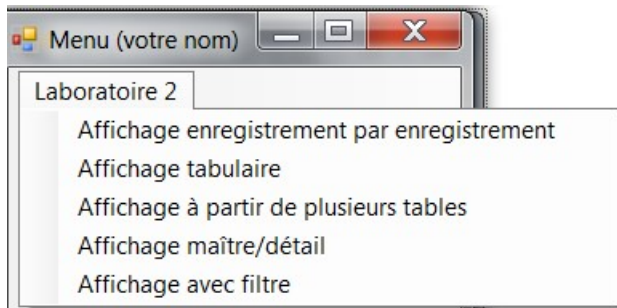
Données	
Connection	BDVoyagesConnectionString (Settings)
DeleteCommand	(DeleteCommand)
GenerateDBDirectMethods	True
InsertCommand	(InsertCommand)
SelectCommand	(SelectCommand)
UpdateCommand	(UpdateCommand)
Génération de code	
BaseClass	System.ComponentModel.Component
ConnectionModifier	Internal
Modifier	Public
Name	contratTableAdapter

Notez la présence des propriétés *DeleteCommand*, *InsertCommand*, *SelectCommand* et *UpdateCommand*.

La propriété **Connection** contient votre chaîne de connexion.

4. Ajout d'un menu sur le formulaire qui servira à appeler les formulaires des sections 5, 6, 7, 8 et 9

Sur votre formulaire de départ (renommez-le **frmMenu**), ajoutez le menu (un *MenuStrip*) suivant :



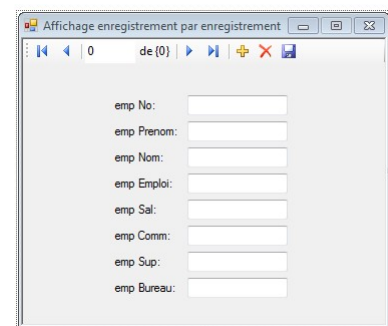
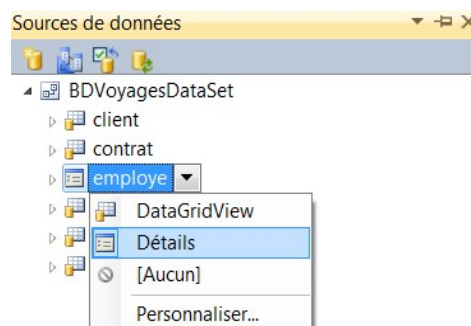
N'oubliez pas d'écrire votre nom dans le titre du formulaire.

5. Gestion des employés enregistrement par enregistrement

Dans cette application, un seul enregistrement à la fois sera affiché dans des zones de texte d'un formulaire.

Étapes :

- Créez un nouveau formulaire nommé **frmEnregistrement** puis fixez sa propriété *FormBorderStyle* à *Fixed3D* et sa propriété *MaximizeBox* à *False*. Le titre du formulaire doit être **Affichage enregistrement par enregistrement**.
- Affichez la fenêtre *Sources de données* et développez le nœud *BDVoyagesVotreNomDataSet* si nécessaire.
- Développez la liste de choix de la *DataTable* **employe** et sélectionnez *Détails* comme suit :



- Glissez la *DataTable* **employe** de la fenêtre *Sources de données* vers le formulaire. Repositionnez éventuellement les contrôles.
- Sur le formulaire, ajoutez une étiquette *Gestion des employés* et modifiez le texte des étiquettes (associées aux zones de texte) pour que celles-ci soient plus lisibles pour l'utilisateur.

- Rattachez-le à *frmMenu*. Pour ce faire, complétez le code de la classe *frmMenu* comme suit :

```
namespace Labo2
{
    public partial class frmMenu : Form
    {
        frmEnregistrement fEnr = new frmEnregistrement(); // Code à ajouter

        public frmMenu()
        {
            InitializeComponent();
        }

        private void affichageEnregistrementParEnregistrementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Code à ajouter
            fEnr.ShowDialog(); // Code à ajouter
            this.Show(); // Code à ajouter
        }
    }
}
```

Double-cliquez sur l'item **Affichage enregistrement par enregistrement** pour créer cet événement dans la fenêtre de code.

Affichage enregistrement par enregistrement

- Exécutez l'application.
- Utilisez la barre de navigation pour parcourir tous les enregistrements de la *DataTable* **Employe**.




- Vous pouvez aussi ajouter, supprimer un enregistrement ainsi qu'enregistrer les modifications à l'aide de cette barre de navigation. Ne le faites pour l'instant.
- **Attention** : L'enregistrement des modifications est nécessaire si on souhaite les sauvegarder dans la vraie base de données.

Pour illustrer cela, modifiez le nom de *Martine Lachance* (employé 1005) par *Josée Lachance*.

Mettez fin à l'exécution de l'application sans enregistrer cette modification puis exécutez l'application de nouveau.

Observez que le prénom de *Martine Lachance* n'a pas été modifié.

Numéro:	1005
Prénom:	Martine
Nom:	Lachance

Faites la même modification mais, cette fois, enregistrez la modification  puis exécutez l'application de nouveau.

Observez que le prénom de *Martine Lachance* a été modifié.

Observez l'événement **Load** du formulaire **frmEnregistrement**. Ce code a été généré automatiquement.

```
private void frmEnregistrement_Load(object sender, EventArgs e)
{
    // TODO: cette ligne de code charge les données dans la table 'bDVoyagesDataSet.employe'.
    this.employeTableAdapter.Fill(this.bDVoyagesDataSet.employe);
}
```

Dans cet événement *Load*, le *TableAdapter* de la table **employe** remplit (méthode **Fill**) la *DataTable* **employe** du *DataSet* et ce, à partir des données de la table **employe** du serveur.

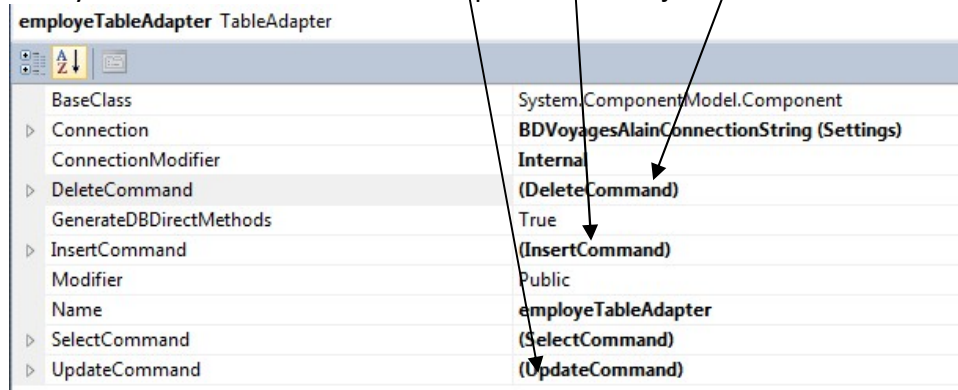
Observez l'événement *Click* associé à l'enregistrement des modifications à partir de la barre de navigation.

```
private void employeBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.employeBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.bDVoyagesDataSet);
}
```

Dans cet événement, le *TableAdapterManager* parcourt, un après l'autre, chacun des *TableAdapter*. Ce dernier met à jour, dans base de données, toutes les données qui ont été modifiées ainsi que tous les enregistrements qui ont été ajoutés ou supprimés à l'intérieur de la *DataTable* correspondante située dans le *DataSet*.

À chaque fois que vous modifiez, ajoutez ou supprimez un enregistrement dans une *DataTable* du *DataSet*, il y a une propriété (*DataRowState*) qui indique le type de mise à jour que vous avez effectué.

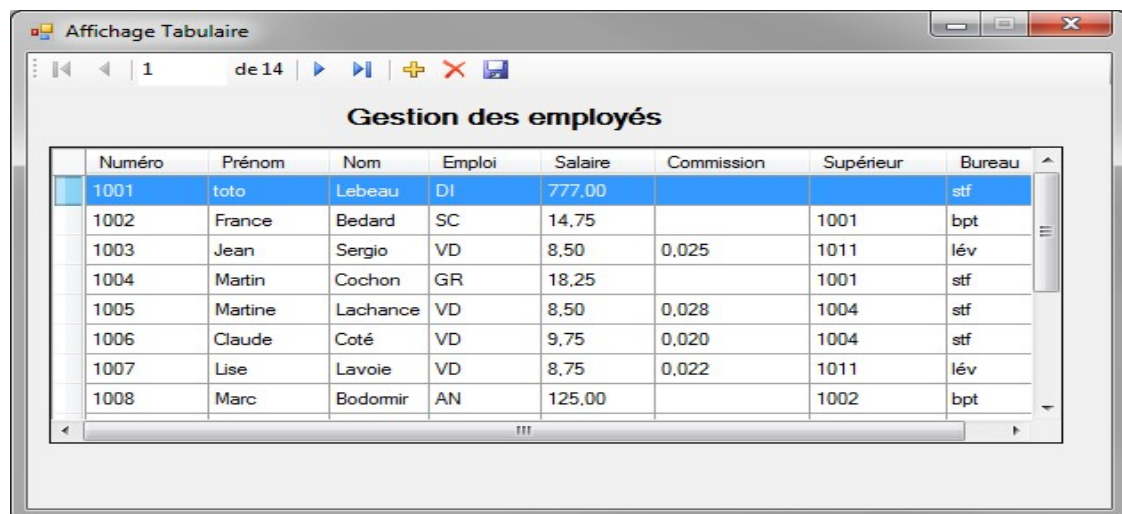
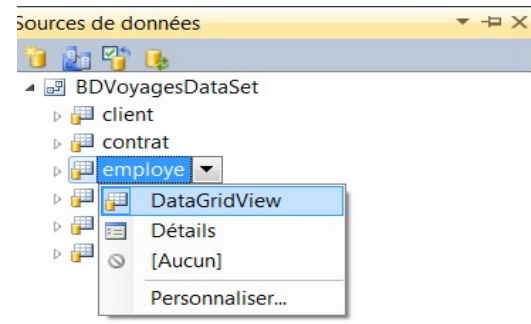
Par conséquent, au moment de l'enregistrement des modifications, le *TableAdapter* utilise la commande SQL (**UPDATE**, **INSERT** ou **DELETE**) correspondante qu'il doit envoyer au fournisseur de données pour mettre à jour les données sur le serveur.



6. Gestion des employés sous forme tabulaire

Refaites toutes les étapes de la section 5, en créant un nouveau formulaire nommé **frmTabulaire** mais en choisissant cette fois-ci le *DataGridView* dans la liste. Le titre du formulaire doit être **Affichage Tabulaire**.

Votre formulaire (en exécution) doit ressembler à celui-ci (il est possible que vos données soient différentes des miennes).



Propriétés de *employeDataGridView* :

Propriété	Nouvelle valeur
<i>Name</i>	<i>dgEmploye</i>
<i>AutoSizeColumnsMode</i>	<i>DisplayedCells</i>
<i>RowHeadersWidth</i>	<i>20</i>
<i>SelectionMode</i>	<i>FullRowSelect</i>

Pour modifier le titre des colonnes, cliquez-droit sur le *DataGridView* puis sélectionnez *Modifier les colonnes* puis modifiez la propriété *HeaderText* de la colonne correspondante.

N'oubliez pas de rattacher ce formulaire à **frmMenu**.

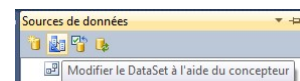
7. Contrats des employés à partir de 2 tables

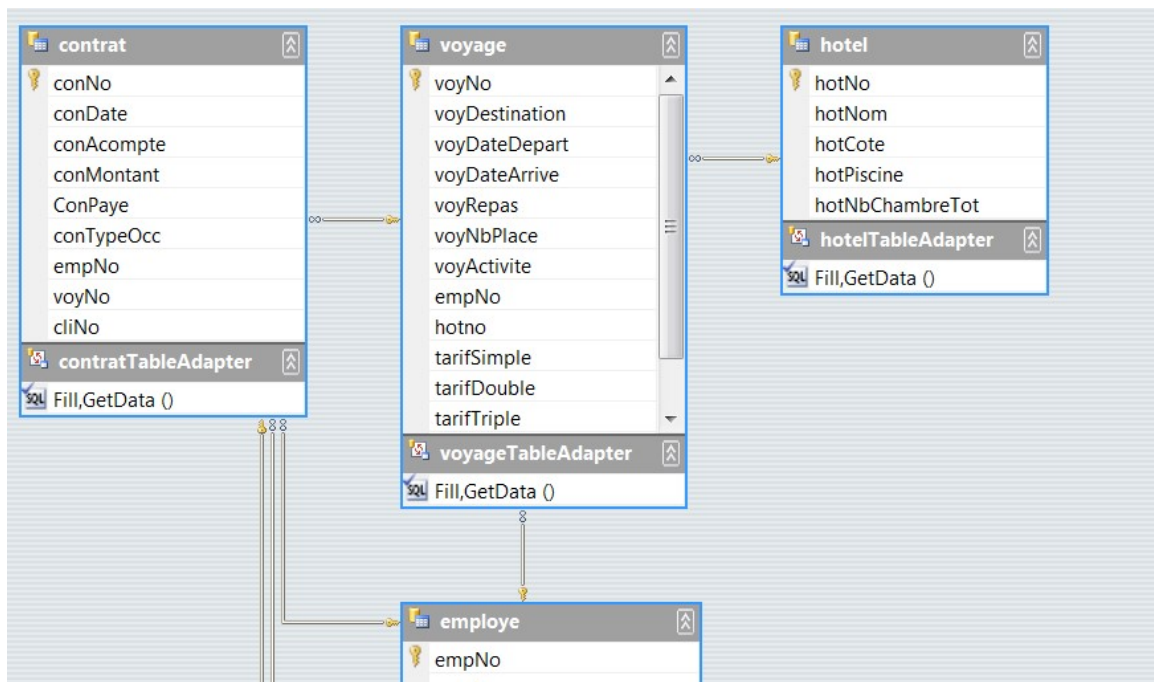
On veut montrer un *DataGridView* qui affiche des informations sur chaque contrat ainsi que le nom et le prénom de l'employé responsable de ce contrat sachant que seul le numéro de l'employé figure dans la table **contrat** et que le nom et le prénom de l'employé figure dans la table **employe**.

Il s'agit donc d'utiliser plus d'une table réelle pour extraire ces informations. Pour implémenter ceci, nous devons, tout d'abord, créer un nouveau *TableAdapter* (ainsi qu'une nouvelle *DataTable*) basé sur une jointure des tables **contrat** et **employe**.

Étapes :

- Créez un formulaire nommé **frmPlusieursTables** puis fixez sa propriété *FormBorderStyle* à *Fixed3D* et sa propriété *MaximizeBox* à *False*. Le titre du formulaire doit être **Affichage à partir de plusieurs tables**.
- Affichez la fenêtre *Sources de données*.
- Dans la fenêtre *Sources de données*, cliquez sur le bouton *Modifier le DataSet à l'aide du concepteur*.
- La fenêtre du concepteur ressemble à ceci :





- Cliquez droit dans un espace libre de ce concepteur, choisissez *Ajouter* puis *TableAdapter* pour ouvrir l'Assistant Configuration de TableAdapter.
- Acceptez la connexion actuelle puis cliquez sur *Suivant*.
- Sur la page *Choisissez un type de commande*, vérifiez que le bouton radio *Utiliser des instructions SQL* est bien sélectionné puis cliquez sur *Suivant*.
- Sur la page *Entrez une instruction SQL*, vous pouvez entrer directement la requête ou bien utiliser le générateur de requêtes. Dans cette section, nous allons utiliser le générateur de requêtes. Cliquez sur *Générateur de requêtes...*
- Dans la boîte de dialogue *Ajouter une table*, sélectionnez les tables **contrat** et **employe**, puis cliquez sur *Ajouter*. Les deux tables devraient être présentes dans la fenêtre *Générateur de Requêtes*. Fermez la boîte de dialogue *Ajouter une table*.
- Dans la fenêtre *Générateur de Requêtes*, cochez les colonnes qu'on veut afficher : *conNo*, *conDate*, *conMontant* pour la table **contrat** et *empNom*, *empPrenom* pour la table **employe**.
- Notez la présence d'une grille de présentation des données ainsi que le corps de la requête SQL générée dans la fenêtre *Générateur de Requêtes*. Ceci est la commande SQL que le *TableAdapter* va envoyer au serveur.

Colonne	Alias	Table	Sortie	Type de tri	Ordre de tri	Filtrer	Ou...
conNo		contrat	<input checked="" type="checkbox"/>				
conDate		contrat	<input checked="" type="checkbox"/>				


```

SELECT  contrat.conNo, contrat.conDate, contrat.conMontant, employe.empNom, employe.empPrenom
FROM    contrat INNER JOIN
        employe ON contrat.empNo = employe.empNo

```

- Complétez la colonne *Alias* (pour que le nom du champ soit plus clair).

Colonne	Alias	Table
conDate	dateContrat	contrat
conMontant	montantContrat	contrat
empPrenom	prenomEmploye	employe
empNom	nomEmploye	employe

- Observez que la requête **SELECT** est modifiée en conséquence.
- On aimerait modifier la requête SQL pour que le prénom et le nom de l'employé soient sur une seule colonne. Pour ce faire, modifiez la requête SQL comme suit :

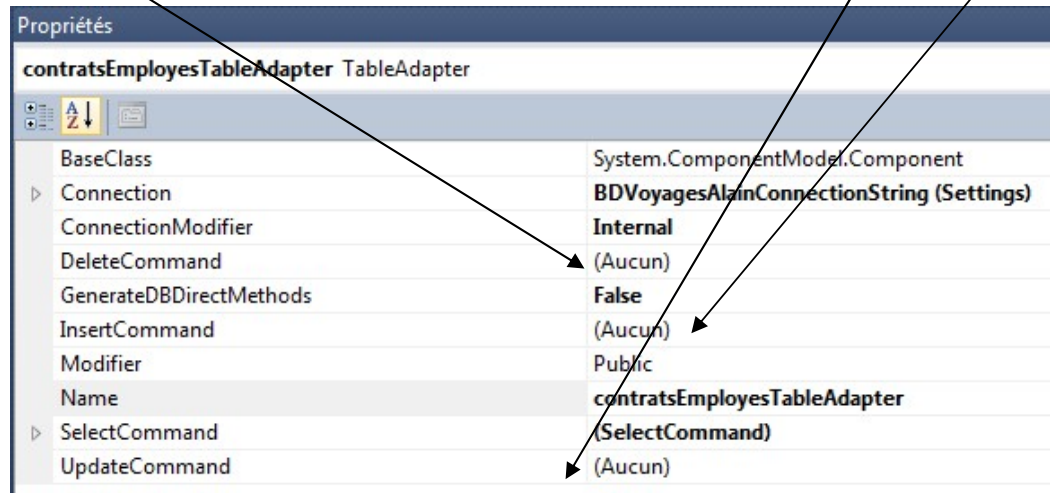
```

SELECT  contrat.conNo AS numeroContrat, contrat.conDate AS dateContrat,
        contrat.conMontant AS montantContrat, employe.empPrenom + ' ' + employe.empNom AS
        prenomEtNomEmploye
FROM    contrat INNER JOIN
        employe ON contrat.empNo = employe.empNo

```

- Pour avoir un aperçu des résultats de la requête, cliquez sur le bouton *Exécuter la requête*.
- Cliquez sur **OK**. L'*Assistant Configuration de TableAdapter* réaffiche la requête modifiée. Cliquez sur *Suivant*.
- Sur la page *Choisir les méthodes à générer*, cliquez sur *Suivant* sans rien modifier.
- Sur la page *Résultats de l'Assistant*, notez les avertissements. Les requêtes **INSERT**, **UPDATE** et **DELETE** ne seront pas générées par le *TableAdapter*. C'est normal, car ces requêtes sont automatiquement générées uniquement dans le cas d'une *DataTable* basée sur une seule table réelle. On ne peut pas faire d'insertion, ni de mise à jour ni de suppression lorsqu'il y a une jointure.
- Cliquez sur *Terminer*. Une nouvelle *DataTable* () et son *TableAdapter* ont été créés.

- Dans la fenêtre *Sources de données*, remarquez la présence d'une nouvelle *DataTable*.
- Dans la fenêtre du concepteur, renommez la *nouvelle DataTable* en **contratsEmployes**.
- Observez, dans les propriétés du *TableAdapter*, aucune requête **UPDATE**, **INSERT** et **DELETE** n'a été automatiquement générée.



- Refaites les étapes de la section 5 mais assurez-vous que vos données sont situées dans un *DataGridView*.

Affichage à partir de plusieurs tables

1 de 18

	conNo	dateContrat	montantContrat	PrenomEtNomEm
▶	444001	2019-05-04	1140,58	Claude Côté
	444002	2019-05-12	725,95	Jean Sergio
	444003	2019-05-12	2028,08	Jean Sergio
	444004	2019-05-26	343,21	Martine Lachance
	444005	2019-05-24	132,90	Jean Sergio
	444006	2019-02-12	3184,84	Lise Lavoie
	444007	2019-02-23	2308,89	Claude Côté
	444008	2019-05-07	779,70	Claude Côté
	444009	2019-04-04	1100,00	Claude Côté




Propriétés du *DataGridView* :

Propriété	Nouvelle valeur
<i>Name</i>	<i>dgContratsEmployes</i>
<i>AutoSizeColumnsMode</i>	<i>DisplayedCells</i>
<i>RowHeadersWidth</i>	20
<i>SelectionMode</i>	<i>FullRowSelect</i>

Pour modifier le titre d'une colonne, cliquez droit sur le *DataGridView* puis sélectionnez *Modifier les colonnes* puis modifiez la propriété *HeaderText* de la colonne correspondante.

N'oubliez pas de rattacher ce formulaire à **frmMenu**.

Exécutez puis observez qu'il est impossible, à l'intérieur de ce *DataGridView*, de modifier le nom complet de l'employé. C'est normal. Le nom complet de l'employé provient de deux champs différents (concaténation des champs *empPrenom* et *empNom*).

Par contre, à l'intérieur de ce *DataGridView*, vous pouvez modifier les autres colonnes. Vous pouvez également insérer un nouvel enregistrement () ou en supprimer un () mais vous ne pouvez pas enregistrer ces modifications dans la base de données réelle ().



8. Contrats de chaque employé : affichage Maître/Détail

On veut montrer un *DataGridView* qui affiche des informations sur tous les contrats d'un employé spécifique.

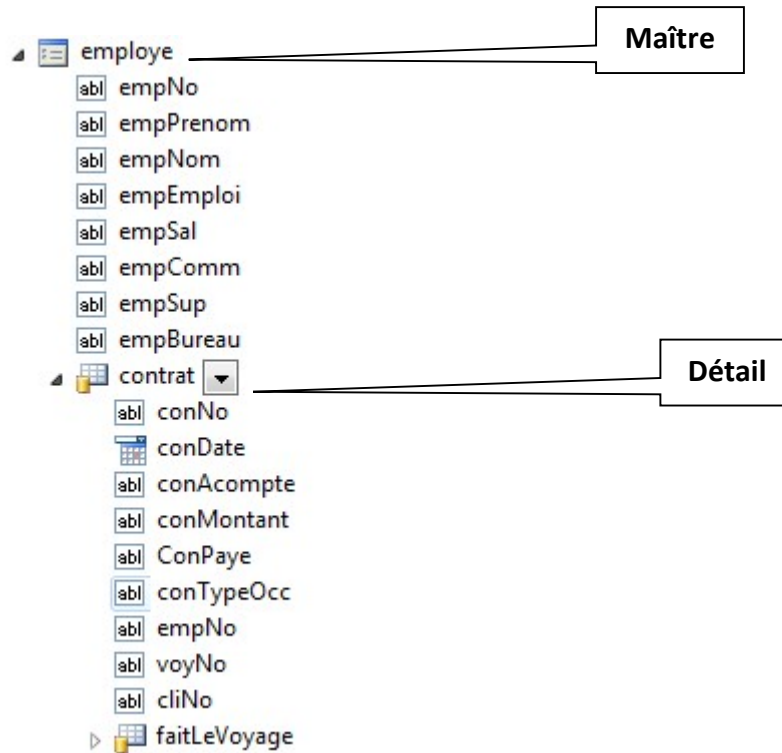
Nous allons procéder de la manière suivante : l'utilisateur va sélectionner un employé spécifique et, à partir du numéro d'employé, tous les contrats de cet employé seront affichés.

Étapes :

- Créez un formulaire nommé **frmMaitreDetail** puis fixez sa propriété *FormBorderStyle* à *Fixed3D* et sa propriété *MaximizeBox* à *False*. Le titre du formulaire doit être **Affichage maître/détail**.
- Sur ce formulaire, affichez la table **employe** enregistrement par enregistrement à la manière de la section 5. Affichez seulement le numéro, le prénom et le nom (en supprimant les contrôles qui ne sont pas nécessaires) puis disposez-les de la manière suivante :

Numéro:
 Prénom:
 Nom:

- Maintenant, affichez les contrats sous forme tabulaire à la manière de la section 6 dans un *DataGridView*. Notez qu'il faut choisir la *DataTable* **contrat** qui apparaît dans l'arborescence de la *DataTable* **employe** comme l'illustre la figure suivante :



- Dans le *DataGridView*, n'affichez que les colonnes qui contiennent le numéro de contrat (*conNo*), la date du contrat (*conDate*), le montant du contrat (*conMontant*) et le numéro d'employé (*empNo*).

Pour supprimer des colonnes, cliquez-droit sur le *DataGridView* puis sélectionnez *Modifier les colonnes* puis supprimez les colonnes que vous ne désirez pas.

- Modifiez les titres des colonnes restantes de la manière suivante :

	No Contrat	Date	Montant	No Employe
--	---------------	------	---------	---------------

Propriétés du *DataGridView* :

Propriété	Nouvelle valeur
<i>Name</i>	<i>dgContrats</i>
<i>AutoSizeColumnsMode</i>	<i>Fill</i>
<i>SelectionMode</i>	<i>FullRowSelect</i>

- Ajoutez une étiquette **Contrats** à la gauche du *DataGridView*.
- N'oubliez pas de rattacher ce formulaire à **frmMenu**.
- Voici à quoi devrait ressembler votre formulaire (en exécution).

A screenshot of a Windows application window titled "Affichage maître/détail". The window contains a master-detail form. At the top, there are three text boxes: "Numéro: 1003", "Prénom: Jean", and "Nom: Sergio". Below these is a table labeled "Contrats". The table has four columns: "No Contrat", "Date", "Montant", and "No Employe". The table contains six rows of data, with the first row highlighted in blue. The data in the table is as follows:

No Contrat	Date	Montant	No Employe
444002	2019-05-12	725,95	1003
444003	2019-05-12	2028,08	1003
444005	2019-05-24	132,90	1003
444011	2018-12-27	501,40	1003
444016	2019-05-02	3000,00	1003
444018	2018-12-10	342,70	1003

Observez ici que les contrats affichés dans le *DataGridView* dépendent du numéro d'employé. Cela est possible grâce à la relation qui existe entre le numéro d'employé dans la table **employe** (la clef primaire) et le numéro d'employé dans la table **contrat** (la clef étrangère).

9. Affichage à partir d'un filtre

Si on veut créer un formulaire qui affiche les enregistrements qui vérifient un critère de recherche, nous devons créer un objet *TableAdapter* qui va utiliser un paramètre *SQL Serveur*.

Dans notre cas, nous allons afficher les noms des employés qui contiennent une chaîne de caractères donnée.

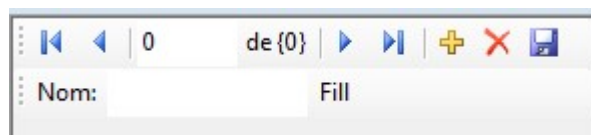
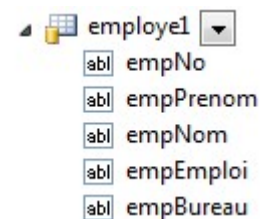
Étapes :

- Créez un formulaire nommé **frmAffichageAvecFiltre** puis fixez sa propriété *FormBorderStyle* à *Fixed3D* et sa propriété *MaximizeBox* à *False*. Le titre du formulaire doit être **Affichage avec filtre**.
- Créez un nouveau *TableAdapter* (ainsi qu'une nouvelle *DataTable*) de la même manière qu'à la section 7 mais en utilisant que la table **employe**. Sur la page *Entrez une instruction SQL*, tapez la requête suivante puis cliquez sur le bouton *Terminer*.

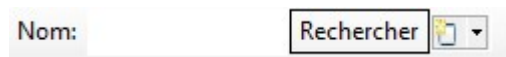
```
SELECT empNo, empPrenom, empNom, empEmploi, empBureau FROM employe  
WHERE empNom LIKE '%' + @nom + '%'
```

Noter que **@nom** est un paramètre *SQL Serveur* (voir laboratoire 1).

- Dans la fenêtre *Sources de données*, remarquez la présence de la *DataTable* **employe1**.
- Donnez-lui le nom de **employesAvecFiltre**.
- Dans cette fenêtre, utilisez cette *DataTable* pour un affichage tabulaire sur le formulaire *frmAffichageAvecFiltre*.



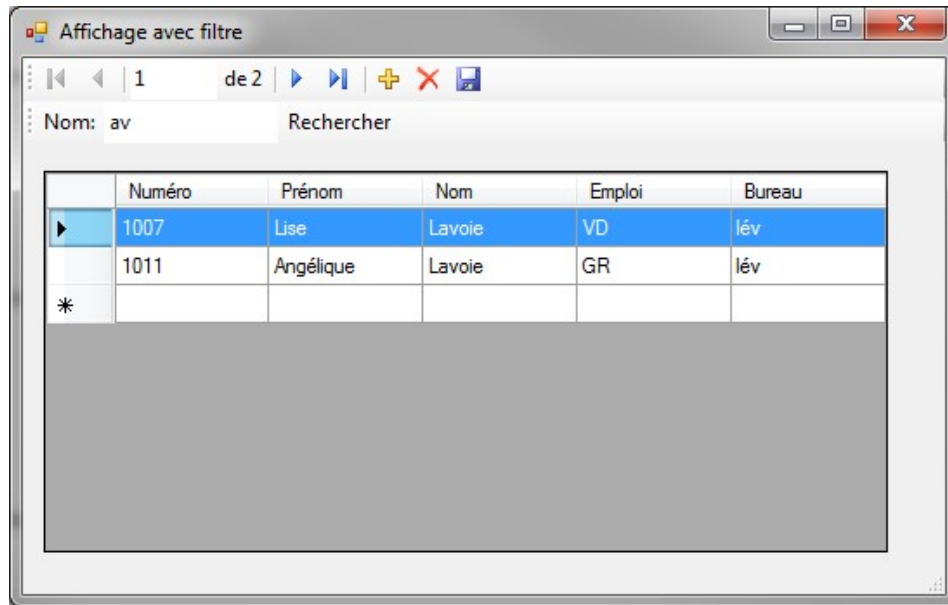
- Changez le mot *Fill*. Dans la propriété *Text*, tapez *Rechercher* au lieu de *Fill*.



- Renommez les colonnes et modifiez les propriétés du *DataGridView* pour avoir ce qui suit :

	Propriété	Nouvelle valeur
N	Name	dgEmployesAvecFiltre
'	AutoSizeColumnMode	Fill
o	SelectionMode	FullRowSelect

- N'oubliez pas de rattacher ce formulaire à **frmMenu**.
- Exécutez en entrant une chaîne de caractères dans la zone de texte *Nom* puis cliquez sur *Rechercher* pour voir les résultats.



Il est à noter que ce que vous tapez dans la zone de texte **Nom** est la valeur du paramètre SQL Serveur (*@Nom*) que le *TableAdapter* transmet à la requête SQL.