

Laboratoire 5
Utilisation de listes de valeurs
Ajout de champs calculés à une *DataTable*

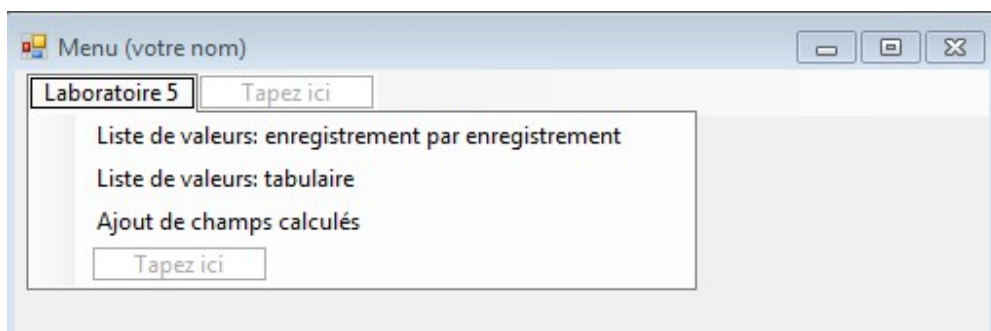
1. Objectifs d'apprentissage

- Utiliser des listes de valeurs
- Ajouter de champs calculés à une *DataTable*
- Styliser les colonnes d'un DataGridView

2. Création du formulaire frmMenu

Pour retrouver les données initiales de votre base de données, exécutez les scripts *BDVoyages.sql* et *DonneesBDVoyages.sql*.

Dans un nouveau projet nommé **Labo5**, sur le formulaire **frmMenu**, créer un menu (un *MenuStrip*) pour rattacher les formulaires des sections 3, 4 et 5.



Le titre du formulaire doit être *Menu (votre nom)*.

3. Création d'une liste de valeurs : cas affichage enregistrement par enregistrement

Les listes de valeurs sont utilisées pour éviter les erreurs d'intégrité référentielle lors d'insertion de données. Par exemple, si on veut ajouter un nouveau contrat, le numéro de voyage, le numéro de client et le numéro d'employé sont des clés étrangères. Nous pouvons donc choisir ces valeurs à partir de listes qui contiennent les valeurs présentes dans la base de données.

Il est important de noter qu'on peut faire afficher le nom d'un employé au lieu de son numéro même si c'est le numéro qui sera inséré dans la base de données.

Étapes à suivre :

- À l'aide de l'assistant, créez un *DataSet* pour la base de données **BDVoyagesVotreNom** (utilisez une connexion avec une authentification *SQL Serveur*). Toutes les tables doivent être présentes.
- Créez un nouveau formulaire **frmListeDeValeurs** dont le but sera d'afficher les enregistrements de la table **contrat** enregistrement par enregistrement. Son titre doit être *Liste de valeurs : enregistrement par enregistrement*.
- Reliez **frmListeDeValeurs** à **frmMenu**.

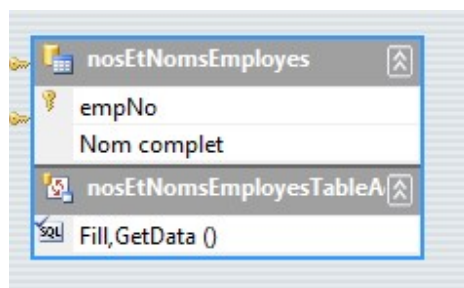
- À l'aide du concepteur de *DataSet*, ajoutez un *TableAdapter* qui utilise la requête suivante.

Quelles données doivent être chargées dans la table ?

```
SELECT empNo, empNom + ' ' + empPrenom AS "Nom complet"
FROM employe
ORDER BY empNom, empPrenom
```

- Renommez le *DataTable*, **nosEtNomsEmployes**.

- À partir de la fenêtre *Sources de données*, faites glisser la *DataTable* **contrat** sur votre formulaire en s'assurant de choisir *Détails*.



- Dans la 7^{ème} étiquette, remplacez le texte *emp No* par *Employé*.

- Supprimez la zone de texte située à droite de l'étiquette *employé* puis mettez une liste combinée (un *ComboBox*) à sa place.

con Type Occ:

employé:

voy No:

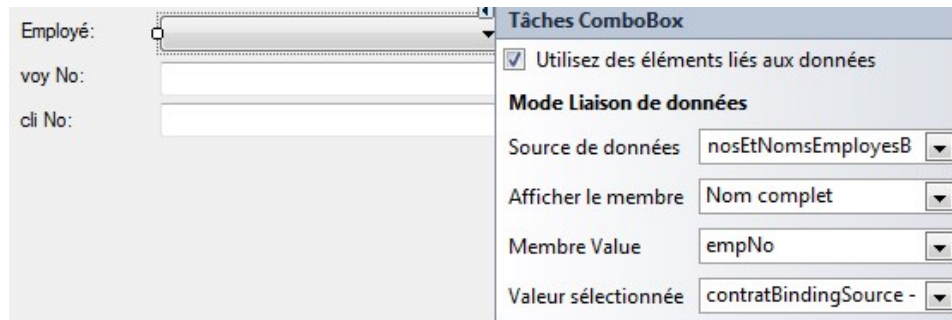
- Changez les propriétés de la liste combinée comme suit :

| Propriété | Valeur |
|---------------|--------------|
| Name | cboEmployes |
| DropDownStyle | DropDownList |

La valeur *DropDownList* permet d'avoir un accès en lecture seule

- Pour lier la liste combinée à la *DataTable* **nosEtNomsEmployes**, à partir de la fenêtre *Sources de données*, glissez la *DataTable* **nosEtNomsEmployes** sur la liste combinée. Cela configure automatiquement la liste combinée.

- Sélectionnez la liste combinée, ouvrez sa liste de tâches en cliquant sur la petite flèche au coin supérieur droit.



- Mettez à jour les propriétés suivantes :

| Propriété | Valeur |
|---------------------|--------------------------------|
| Source de données | nosEtNomsEmployesBindingSource |
| Afficher le membre | Nom complet |
| Membre value | empNo |
| Valeur sélectionnée | contratBindingSource-empNo |

Explications :

La source de données **nosEtNomsEmployesBindingSource** est le contrôleur qui fait le lien entre le modèle (la *DataTable* **nosEtNomsEmployes**) et la vue (la liste combinée **cboEmployes**).

« Afficher le membre » signifie que l'utilisateur va voir le nom complet des employés dans la liste combinée.

« Membre value » signifie que, lorsque l'utilisateur va sélectionner le nom complet, c'est la valeur du champ **empNo** qui va être retournée.

« Valeur sélectionnée » signifie que cette valeur sélectionnée va être affectée à ce champ.

Par exemple, ici, si l'utilisateur sélectionne l'employé *Bedard, France*, la valeur **1002** va être affectée au champ **empNo** de la *DataTable* **contrat** (c'est le contrôleur **contratBindingSource** qui effectue cette opération).

L'inverse est également vrai. Si, dans la *DataTable* **contrat**, le champ **empNo** vaut **1002**, c'est *Bedard, France* qui va être automatiquement sélectionnée dans la liste.

- **TRÈS IMPORTANT** : Lorsque vous créez une *DataTable* dans le but de la rattacher à une liste de valeurs, le nombre de champs doit être de deux exactement. Le

premier champ doit contenir la valeur sélectionnée et le deuxième champ doit contenir ce que l'utilisateur va voir dans la liste.

- Testez l'exécution (qui devrait ressembler à ceci).

Noter que vous pouvez sélectionner l'employé que vous voulez en utilisant son nom complet et non pas son numéro. Par contre, lorsque vous sélectionnez un nom, c'est son numéro qui est retourné.

Employé: Coté, Claude
voy No:
cli No:

- Bedard, France
- Bodomin, Marc
- Bout, Ti
- Cochon, Martin
- Coté, Claude
- Côté, Paul
- DeGuillemets, Virgule
- Lachance, Martine
- Lavoie, Angélique
- Lavoie, Lise
- Lebeau, Marcel
- Lebeau, toto
- Sergio, Jean
- Vitro, Angèle

- Compléter l'application, en remplaçant le numéro de voyage et le numéro de client par des listes de valeurs.

Pour les voyages, affichez, dans la liste combinée, les numéros de voyage suivis de la destination entre parenthèses. Ces éléments doivent être ordonnés par destination puis par numéro de voyage. Le nom de la *DataTable* doit être **nosEtDestinationsVoyages**. Le nom de la liste combinée doit être **cboVoyages** et son style doit être *DropDownList*.

Employé: Coté, Claude
Voyage: 3004 (Tadoussac)
cli No:

- 3001 (Boston)
- 3003 (Boston)
- 3008 (Boston)
- 3009 (Boston)
- 3002 (Montréal)
- 3006 (Montréal)
- 3007 (Montréal)
- 3004 (Tadoussac)
- 3005 (Tadoussac)
- 3010 (Toronto)

Pour les clients, affichez, dans la liste combinée, les noms complets des clients (de la même façon que pour les employés). Ces éléments doivent être ordonnés par nom de famille puis par prénom. Le nom de la *DataTable* doit être **nosEtNomsClients**. Le nom de la liste combinée doit être **cboClients** et son style doit être *DropDownList*.

Voyage: 3004 (Tadoussac)
Client: Sefrest, Wallace

- Brown, Charlie
- Cailloux, Délima
- Cailloux, Fred
- Castafiore, Bianca
- Haddock, Capitaine
- Ketchup, Red
- Lagaffe, Gaston
- Laroche, Arthur
- Laroche, Bertha

- Modifiez les étiquettes comme suit :

Liste de valeurs : enregistrement par enregistrement

No de contrat: 444001
Date: 4 mai 2019
Acompte: 200.00
Montant: 1140.58
Montant payé: 200.00
No places: 3
Employé: Coté, Claude
Voyage: 3004 (Tadoussac)
Client: Sefrest, Wallace

Liste de valeurs tabulaire

| No de contrat | Date | Acompte | Montant | Montant payé | Occupants | Employé |
|---------------|------------|---------|---------|--------------|-----------|-------------------|
| 444001 | 2019-05-04 | 200,00 | 1140,58 | 200,00 | 3 | Côté, Claude |
| 444002 | 2019-05-12 | 100,00 | 725,95 | 625,95 | 2 | Sergio, Jean |
| 444003 | 2019-05-12 | 300,00 | 2028,08 | 1300,00 | 2 | Sergio, Jean |
| 444004 | 2019-05-26 | 50,00 | 343,21 | 50,00 | 3 | Lachance, Martine |
| 444005 | 2019-05-24 | 20,00 | 132,90 | 20,00 | 1 | Sergio, Jean |
| 444006 | 2019-02-12 | 500,00 | 3184,84 | 3184,84 | 4 | Lavoie, Lise |
| 444007 | 2019-02-23 | 350,00 | 2308,89 | 1850,00 | 2 | Côté, Claude |
| 444008 | 2019-05-07 | 117,00 | 779,70 | 179,70 | 2 | Côté, Claude |
| 444009 | 2019-04-04 | 165,00 | 1100,00 | 100,00 | 3 | Côté, Claude |
| 444010 | 2018-11-26 | 50,00 | 342,70 | 200,00 | 2 | Lachance, Martine |
| 444011 | 2018-12-27 | 75,00 | 501,40 | 201,40 | 4 | Sergio, Jean |
| 444012 | 2018-12-27 | 50,00 | 342,70 | 142,70 | 2 | Lachance, Martine |
| 444013 | 2018-12-12 | 342,70 | 342,70 | 0,00 | 2 | Lavoie, Lise |
| 444014 | 2018-11-04 | 100,00 | 667,00 | 567,00 | 4 | Côté, Claude |
| 444015 | 2019-03-20 | 300,00 | 1758,35 | 758,35 | 1 | Côté, Claude |
| 444016 | 2019-05-02 | 450,00 | 3000,00 | 2800,00 | 2 | Sergio, Jean |
| 444017 | 2019-01-02 | 363,80 | 2425,35 | 363,80 | 1 | Côté, Claude |
| 444018 | 2018-12-10 | 50,00 | 342,70 | 142,70 | 2 | Sergio, Jean |

- En procédant de la même manière, complétez l'application en remplaçant le numéro de voyage et le numéro de client par des listes de valeurs. Utilisez la *DataTable* **nosEtDestinationsVoyages** pour la colonne des voyages et la *DataTable* **nosEtNomsClients** pour la colonne des clients.

5. Ajout de champs (colonnes) calculés à une *DataTable*

Il est possible d'ajouter des champs à une *DataTable* déjà existante.

Ces champs contiennent des données qui sont calculées à partir de données de la base.

Étapes :

- Créez un nouveau formulaire **frmChampsCalcules**, Son titre doit être *Ajout de champs calculés*.
- Rattachez ce formulaire à **frmMenu**.
- Reprenez les étapes du laboratoire 2, section 8 (en ajoutant la commission du vendeur) pour obtenir le formulaire maître/détail suivant :

The screenshot shows a Windows form titled "Ajout de champs calculés". It has a toolbar with navigation and action icons. Below the toolbar, there are four input fields: "Numéro:" with the value "1003", "Prénom:" with the value "Jean", "Commission:" with the value "0,025", and "Nom:" with the value "Sergio". Below these fields is a section titled "Contrats" containing a table with 5 columns: "Numéro", "Date", "Montant", and "Numéro d'employé". The table has 7 rows of data, with the first row highlighted in blue.

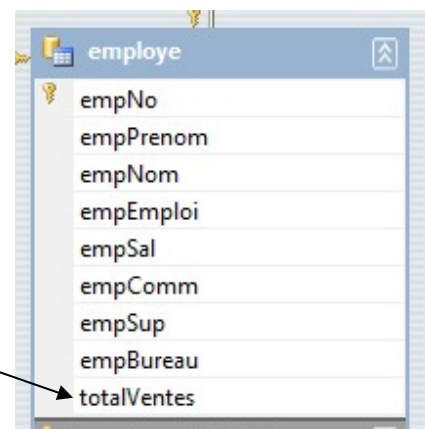
| | Numéro | Date | Montant | Numéro d'employé |
|---|--------|------------|---------|------------------|
| ▶ | 444002 | 2019-05-12 | 725,95 | 1003 |
| | 444003 | 2019-05-12 | 2028,08 | 1003 |
| | 444005 | 2019-05-24 | 132,90 | 1003 |
| | 444011 | 2018-12-27 | 501,40 | 1003 |
| | 444016 | 2019-05-02 | 3000,00 | 1003 |
| | 444018 | 2018-12-10 | 342,70 | 1003 |

Les propriétés du DataGridView sont définies comme suit :

| Propriété | Valeur |
|--------------------------------------|--|
| <i>AllowUserToAddRows</i> | <i>False</i> |
| <i>AllowUserToDeleteRows</i> | <i>False</i> |
| <i>AutoSizeColumnsMode</i> | <i>Fill</i> |
| <i>ColumnHeadersDefaultCellStyle</i> | <i>Font Bold: True</i> <i>Alignment: MiddleCenter</i> |
| <i>ReadOnly</i> | <i>True</i> |
| <i>RowHeadersWidth</i> | <i>40</i> |
| <i>SelectionMode</i> | <i>FullRowSelect</i> |

Utilisez la propriété **DefaultCellStyle** des colonnes pour définir l'alignement et le format **qui respectent l'affichage ci-dessus**.

- A l'intérieur du concepteur du *DataSet*, cliquez droit sur la *DataTable* **employe** et choisir *Ajouter* puis *Colonne* puis donnez le nom de *totalVentes* à cette colonne.

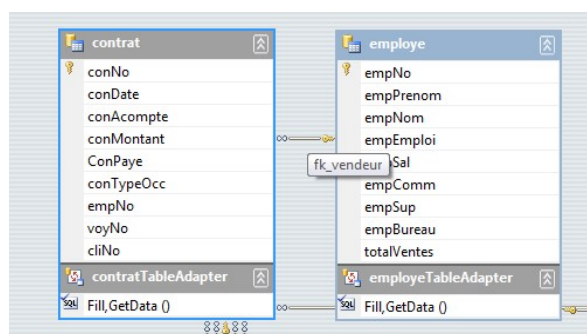


- Définissez les propriétés de cette colonne de la manière suivante :

| Propriété | Valeur |
|------------|-----------------------------------|
| Caption | Total des ventes |
| DataType | System.Decimal |
| Expression | sum(Child(fk_vendeur).conMontant) |

L'*expression* calcule la somme de tous les montants (*conMontant*) des contrats (de l'employé en cours).

Ici, *fk_vendeur* est le nom de la relation dans le *DataSet* qui relie la *DataTable* **employe** à la *DataTable* **contrat**.



- À partir de la fenêtre *Sources de données*, faites glisser, sur le formulaire, la colonne *totalVentes*.

Total des ventes:

- Testez votre application pour différents employés.

- Procédez de la même manière pour faire ce qui suit :
 - Ajoutez une colonne *nbContrats* pour calculer le nombre de contrats. La valeur de la propriété *Expression* est : *count(Child(fk_vendeur).conNo)* et le *Caption* est *Nombre de contrats*.
 - Ajoutez une colonne *plusPetiteVente* pour calculer le montant de la plus petite vente. La valeur de la propriété *Expression* est : *min(Child(fk_vendeur).conMontant)* et le *Caption* est *Plus petite vente*.
 - Ajoutez une colonne *plusGrandeVente* pour calculer le montant de la plus grande vente. La valeur de la propriété *Expression* est : *max(Child(fk_vendeur).conMontant)* et le *Caption* est *Plus grande vente*.
 - Ajoutez une colonne *moyenneVentes* pour calculer la moyenne des ventes. La valeur de la propriété *Expression* est : *avg(Child(fk_vendeur).conMontant)* et le *Caption* est *Moyenne des ventes*.
 - Ajoutez une colonne *commissionPercue* pour calculer la commission totale qui a été perçue. La valeur de la propriété *Expression* est : *totalVentes*empComm* et le *Caption* est *Commission perçue*.

- Tester votre application pour différents employés.

Au niveau des zones de texte, la présentation des données n'est pas très belle.

- Pour toutes les zones de texte (sauf celle qui concerne le nombre de ventes), affecter la valeur *Right* à la propriété *TextAlign*.
- Dans une zone de texte, il n'existe pas de propriété qui donne la possibilité au programmeur de formater automatiquement une donnée (comme c'est le cas pour les données d'un *DataGridView*).

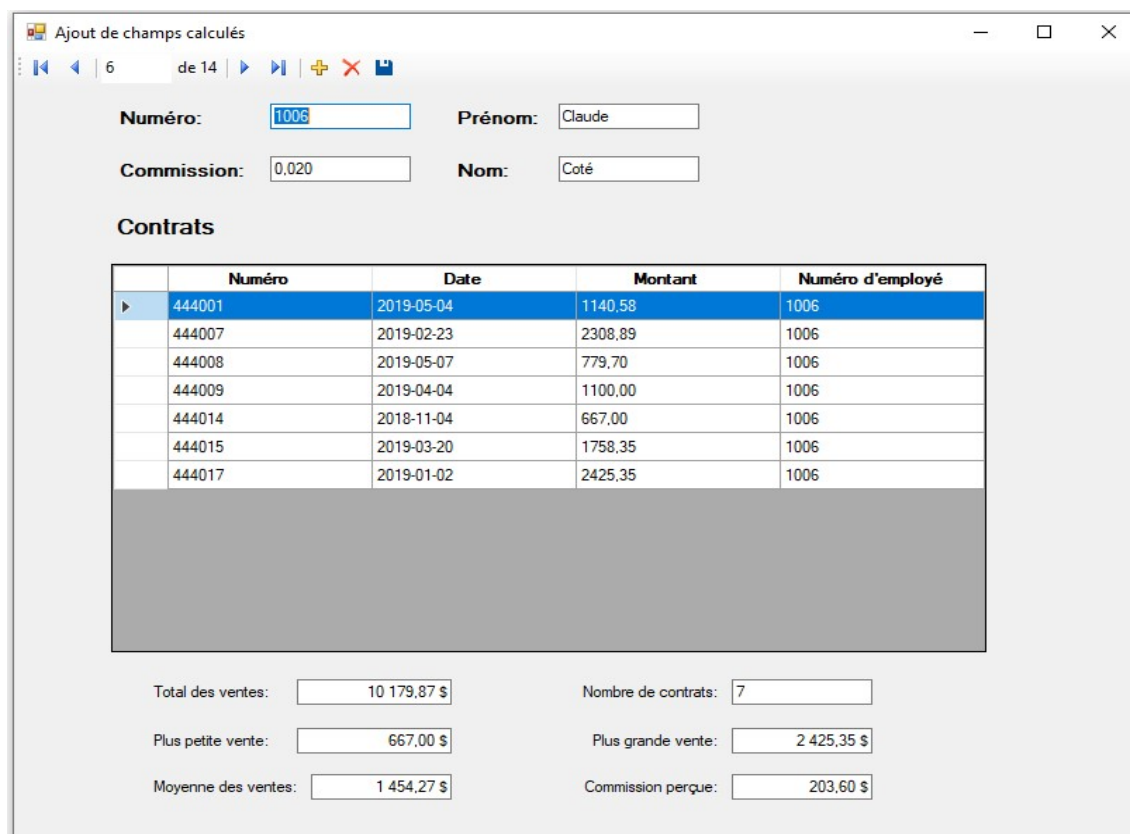
Nous allons formater la donnée par programmation. À chaque fois que le contrôleur va modifier la donnée située dans la zone de texte, nous allons la modifier de nouveau mais, cette fois-ci, en la formatant.

- Dans la fenêtre des propriétés de la zone de texte **totalVentesTextBox**, affecter la valeur *N/A* à la propriété *Text*.
- Programmez l'événement *TextChanged* de la zone de texte **totalVentesTextBox** comme suit :


```
private void totalVentesTextBox_TextChanged(object sender, EventArgs e)
{
    string format = totalVentesTextBox.Text;
    if (format == string.Empty)
    {
        format = "N/A";
    }
    else if (format != "N/A" && format.Substring(format.Length - 1, 1) != "$")
    {
        double montant = Convert.ToDouble(format);
        format = string.Format("{0:C}", montant);
    }

    totalVentesTextBox.Text = format;
}
```

- Exécutez l'application et vérifiez que le total des ventes contient soit N/A, soit un montant d'argent qui se termine par \$.
- Procédez exactement de la même manière pour les quatre (4) dernières zones de texte et vérifiez l'exécution.



Contrats

| | Numéro | Date | Montant | Numéro d'employé |
|---|--------|------------|---------|------------------|
| ▶ | 444001 | 2019-05-04 | 1140,58 | 1006 |
| | 444007 | 2019-02-23 | 2308,89 | 1006 |
| | 444008 | 2019-05-07 | 779,70 | 1006 |
| | 444009 | 2019-04-04 | 1100,00 | 1006 |
| | 444014 | 2018-11-04 | 667,00 | 1006 |
| | 444015 | 2019-03-20 | 1758,35 | 1006 |
| | 444017 | 2019-01-02 | 2425,35 | 1006 |

Total des ventes: 10 179,87 \$ Nombre de contrats: 7

Plus petite vente: 667,00 \$ Plus grande vente: 2 425,35 \$

Moyenne des ventes: 1 454,27 \$ Commission perçue: 203,60 \$

- Il est important de noter que tous les champs calculés sont présents dans la **DataTable employe** située en mémoire mais ne sont pas présents dans la vraie table **employe** située sur le serveur. Ces champs ne sont donc pas pris en compte par le TableAdapter.