

# Enregistrer des données

---

Les fichiers .....	120
Les bases de données .....	120
Écrire dans un fichier .....	122
Lire un fichier .....	126
Ajouter une base de données au projet .....	131
Afficher les données de la base .....	137
Aller plus loin grâce aux bases de données .....	144

Dans tous les programmes que vous avez réalisés jusque-là, lors de la fermeture du programme, toutes les informations saisies sont effacées, ce qui est gênant. Heureusement, il existe des moyens de sauvegarder des données une fois l'exécution d'un programme terminée. Vous allez voir deux méthodes : la première repose sur des fichiers de sauvegarde ; la deuxième, fondée sur les bases de données, est plus complète mais plus longue à mettre en place.

## 7.1. Les fichiers

Chacune de ses deux méthodes possède ses avantages et ses inconvénients. Dans le cas de la sauvegarde par fichier, il faut prendre le temps de définir un système de lecture et d'écriture, réfléchir à l'ordre dans lequel vous allez stocker des informations dans le fichier et l'ordre dans lequel vous allez les lire. Il est courant d'utiliser cette méthode lorsque les informations à stocker ne sont pas nombreuses. En effet, un fichier contenant seulement quelques lignes prend beaucoup moins de place qu'une base de données.

## Les bases de données

Lorsque les informations sont plus nombreuses, on veut généralement les organiser. Prenons le cas de recettes de cuisine. Supposons que vous disposiez de cinq recettes : un gâteau, un plat, deux salades et un clafoutis. Il est facile de les garder toutes dans un tiroir, et chaque fois que l'envie de cuisiner vous saisit, de prendre celle qui vous intéresse. Supposons maintenant que vous avez réuni cinq cents recettes. Il devient dès lors nécessaire de les ranger dans une sorte de classeur, par thèmes, par types de plats ou encore par pays d'origine. Une base de données est simplement le classeur dans lequel vous allez les organiser. Il s'agit d'un système de regroupement et d'organisation des données à la manière d'une bibliothèque.

Sur une feuille de papier, dessinez un rectangle plus haut que large. En haut, inscrivez "Recettes". Tirez un trait juste en dessous. Divisez le reste du tableau pour avoir deux colonnes sous la ligne "Recettes". Dans la colonne de gauche, inscrivez tout ce qui peut caractériser une recette de cuisine. Dans celle de droite, inscrivez de quels types de données informatiques ces critères relèvent. Par exemple, le nombre de personnes pour lequel est prévue la recette peut être représenté dans un

programme par un nombre entier. Voici un exemple de tableau que vous pourriez obtenir :

**Tableau 7.1 : Réflexion autour des recettes**

Donnée	Type
Nombre de personnes	Entier
Temps de préparation	Entier
Nom de la recette	Chaîne de caractères
Catégorie de plats	Chaîne de caractères
Recette	Texte

Cette représentation va servir à établir ce que sera une table dans la base de données. Une table est une partie de la base qui va contenir des informations. Ce que vous venez de faire s'appelle la modélisation de la base de données. En effet, une base de données est organisée sous forme de tables. Dans la table des recettes par exemple, chaque recette va représenter une entrée. Une fois en mémoire, la table consistera en un tableau dont chaque ligne sera une entrée. Une fois rempli, le tableau se présentera comme ceci pour le programme :

**Tableau 7.2 : Table des recettes**

Nombre de personnes	Temps de préparation	Nom de la recette	Catégorie de plats	Recette
4	30 minutes	Clafoutis	Dessert	Dans un grand plat...
2	2 minutes	Salade verte	Entrée	Dans un petit plat...

En quoi cette organisation sous forme de tableau fait-elle gagner du temps ? Si vous avez deux recettes, cela ne sert à rien car la quantité d'informations n'est pas importante. Si vous avez deux mille recettes à trier, une base de données va vous permettre de retrouver la recette que vous cherchez parmi les deux mille disponibles, et ce en moins d'une seconde, grâce à un langage spécifique de requêtes. Rassurez-vous, il n'est pas question que vous appreniez le langage des bases de données pour le moment. Visual Basic se charge de traduire pour vous. Pour le moment, reprenez ce principe : de manière à optimiser le programme à l'aide d'une base de données, il faut d'abord réfléchir à l'organisation de

la base, puis remplir la base à l'aide de données sous forme de tableau et enfin créer une requête de manière à retrouver rapidement les informations de la base. Si vous vous sentez légèrement perdu, rien de grave, nous éclaircirons tout cela par la pratique dans un instant.

Maintenant que vous connaissez les deux méthodes possibles pour sauvegarder des données, passez à la pratique. Vous allez créer cet utilitaire de gestion de recettes.



REMARQUE

### Rapidité

Les bases de données étant des fichiers organisés, il est souvent plus rapide de rechercher une information dans de telles structures de données que dans un fichier sur un disque.

## Écrire dans un fichier

La première méthode que vous allez mettre en œuvre est la sauvegarde par fichier. Son point commun avec la méthode des bases de données est de demander un instant de réflexion quant à son usage. Dans ce cas, vous allez sauvegarder chaque recette dans un fichier différent.

- 1 Commencez par créer un projet dans Visual Basic.

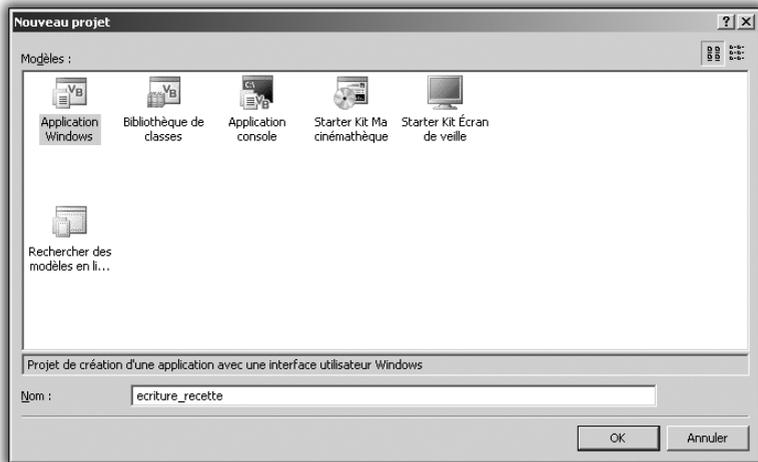


Figure 7.1 : Projet d'écriture de recettes

- 2 Dans le designer de formulaires, ajoutez au formulaire par défaut un contrôle *MenuStrip* puis placez dessus les éléments standard.



*Pour savoir comment procéder, lisez le chapitre **Penser à l'utilisateur**.*



**Figure 7.2 :**  
*Formulaire et menu*

- 3** Dans le formulaire, ajoutez trois *Label* et quatre *TextBox*. Dans la fenêtre des propriétés, changez les propriétés *Name* des *TextBox* en `nombrepers`, `tempsprep`, `titre` et `recette`.

La première va contenir le nombre de personnes pour lequel est prévue la recette, la deuxième le temps de préparation, et les deux autres respectivement le titre et le corps de la recette.



**Figure 7.3 :**  
*Prêt à cuisiner*

Tout n'est pas fini. Pour sauvegarder les données, vous allez utiliser des fichiers contenant du texte sous cette forme :

Nombre de personnes

Temps de préparation

Titre

Corps de la recette sur plusieurs lignes

De cette manière, vous stockerez une information importante par ligne, et il sera aisé de retrouver automatiquement toutes les informations nécessaires quand vous aurez besoin d'ouvrir le fichier contenant la recette.

Il s'agit maintenant que le programme sauvegarde la recette dans un fichier.

#### 4 Ajoutez au formulaire un contrôle de type *SaveFileDialog*.

Dans le designer de formulaires, cliquez sur le menu **Fichier** puis double-cliquez sur le sous-menu **Enregistrer**.

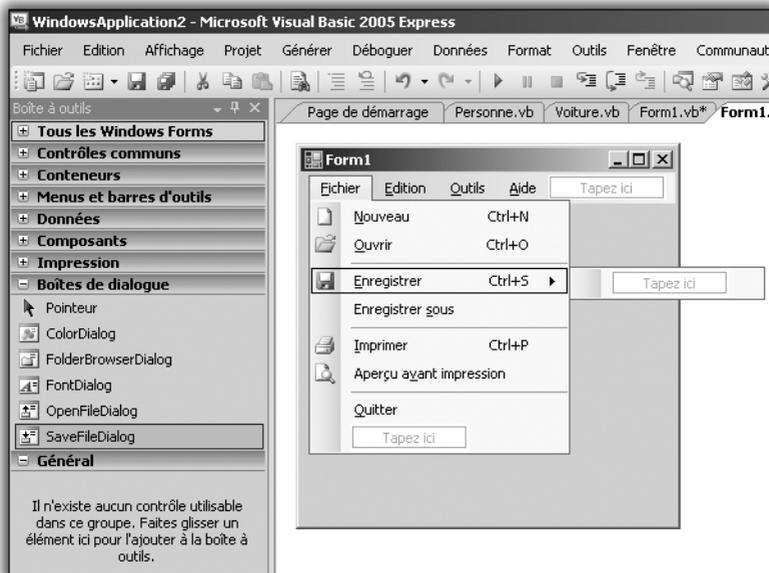


Figure 7.4 : Méthode d'enregistrement

Une fois rendu dans l'éditeur de code, recopiez le code suivant :

```
SaveFileDialog1.Filter = "Fichiers TXT(*.txt)|.txt"
If SaveFileDialog1.ShowDialog = _
    Windows.Forms.DialogResult.OK Then
    Dim sw As _
New System.IO.StreamWriter(SaveFileDialog1.FileName)
    sw.WriteLine(NbPers.Text)
    sw.WriteLine(TempsPrep.Text)
    sw.WriteLine(Titre.Text)
    sw.Write(Recette.Text)
    sw.Flush()
End If
```

Vous précisez en première ligne le type de fichier à sauvegarder. En précisant `TXT`, vous désignez un fichier texte qui sera réutilisable dans n'importe quelle application.

Le test conditionnel permet de vérifier que l'utilisateur a bien cliqué sur le bouton OK de la boîte de dialogue de sauvegarde de fichier. Une fois que vous êtes sûr qu'un fichier a bien été sélectionné, vous exploitez une nouvelle notion, que sont les flux. Un peu à la manière d'un tuyaux, un flux permet de faire passer de l'information d'un point à un autre, comme une rivière permet à un bateau de naviguer d'une ville à un autre. Vous utilisez ici un objet de type `StreamWriter`, que l'on peut traduire par "flux d'écriture". Vous faites passer de l'information d'un point A à un point B en l'écrivant une fois arrivé au point B. Dans ce cas, le point A représente chacune des `TextBox` que vous allez remplir, et le point B représente le fichier d'arrivée dans lequel vous voulez enregistrer la recette.



ATTENTION

### Création du fichier

Lors de la création du fichier, il faut vérifier la place qu'il va prendre sur le disque dur. Si l'espace disque n'est pas suffisant, vous devrez gérer l'erreur.

Vous initialisez le point B (votre fichier) en récupérant dans la boîte de dialogue de sauvegarde le résultat de la sélection de l'utilisateur, en l'occurrence le fichier qu'il a choisi. Ensuite, vous utilisez la méthode `WriteLine` du flux d'écriture pour ajouter, ligne par ligne, le contenu des différentes `TextBox` du formulaire.

La dernière ligne est un peu particulière. Il s'agit en effet de vider complètement le flux d'écriture, de manière à ne perdre aucune

information à l'intérieur. Cette action a pour but de tout faire arriver au point B, un peu comme si l'on essayait de transférer de l'eau d'une bouteille dans un seau. Le flush permet de secouer la bouteille pour ne laisser aucune goutte à l'intérieur. La différence est que le flux d'écriture fonctionne comme une bouteille dans laquelle l'eau peut rester accrochée. Il faut donc secouer pour la faire tomber. Si vous enlevez la ligne `sw.Flush()`, aucune partie du texte ne sera écrite dans le fichier.



REMARQUE

**Flush**

Le « flush » est l'action de vider complètement un flux d'écriture. Cette opération permet réellement de terminer l'écriture de données.

Comme vous pouvez le voir, l'écriture dans un fichier est très simple et se fait grâce à un flux d'écriture.



RENOVI

*Pour rendre cette application plus agréable à utiliser, reprenez les consignes du chapitre **Penser à l'utilisateur en matière de design d'interface**.*

Il faut supprimer les menus inutiles et faire en sorte que le redimensionnement du formulaire principal ne déplace pas les éléments qu'il contient.

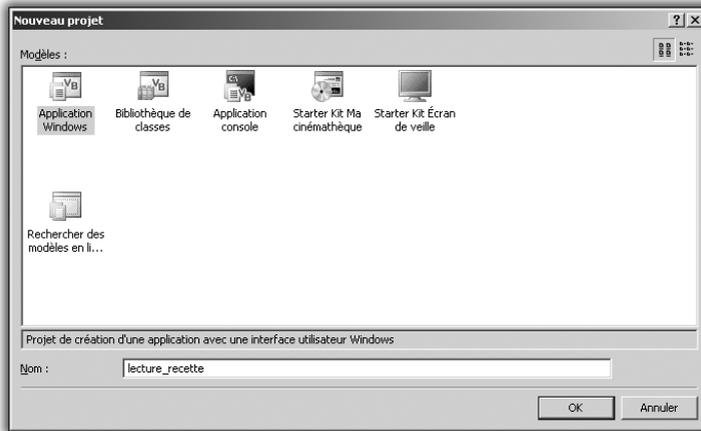
Amusez-vous à entrer quelques recettes de manière à disposer d'un répertoire contenant une dizaine de fichiers.

Passons maintenant à la lecture des fichiers de recette et à leur chargement dans l'application.

## 7.4. Lire un fichier

Votre application permet de sauvegarder des recettes dans un répertoire. Pour que le projet soit complet, il faut maintenant un moyen de récupérer ces informations. Vous allez donc créer un deuxième projet pour lire les recettes dans le répertoire de sauvegarde.

- 1 Créez un nouveau projet de type application Windows et appelez-le `Lecture Recette`.



**Figure 7.5 :** *Projet de lecture des recettes*

- 2 Dans le designer de formulaires, ajoutez au formulaire principal un contrôle *MenuStrip* ainsi que trois contrôles *Label* et une *TextBox*. Dans la fenêtre des propriétés de la *TextBox*, changez la propriété *ReadOnly* en la mettant à la valeur *True*, afin de rendre la *TextBox* non éditable. En effet, vous voulez uniquement lire les informations contenues dans un fichier.



**Figure 7.6 :**  
*Squelette de l'application de lecture*

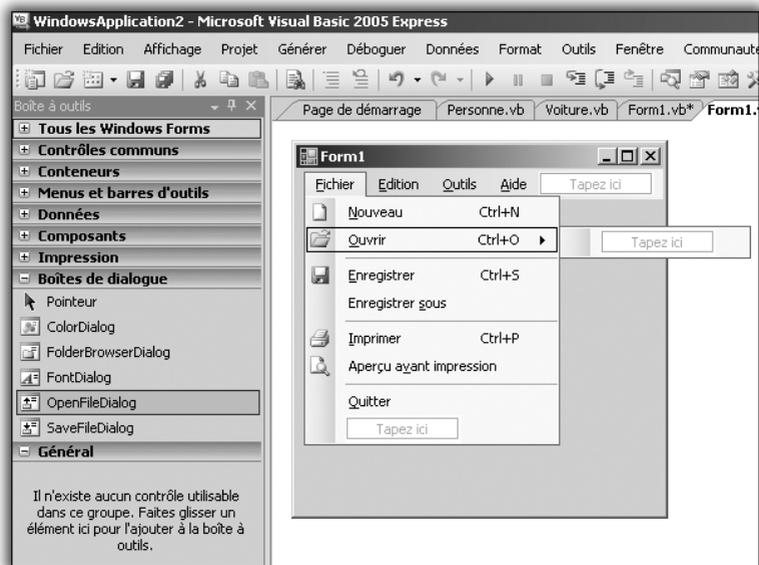
- 3 Maintenant que vous avez le prototype, modifiez le nom des contrôles pour arriver à ce que vous voulez. Les contrôles *Label* vont contenir chacun une des informations de la recette, à savoir le titre, le nombre de personnes pour lequel la recette est prévue et le temps de préparation.
- 4 Ajoutez maintenant un contrôle de type *OpenFileDialog* qui va permettre de sélectionner les recettes.

Il faut passer aux instructions. Comme pour l'écriture dans un fichier, la lecture d'un fichier passe par l'utilisation de flux. La différence cette fois est que le flux va dans l'autre sens, à savoir d'une ressource (votre fichier) vers votre programme. Le fonctionnement est le même, seul le nom va changer. Vous allez en effet utiliser un objet de type `StreamReader`, et non un `StreamWriter`. Dans cet exemple, nous avons renommé les *Label* en fonction des informations qu'ils allaient recevoir.



**Figure 7.7 :**  
*L'interface finale*

- 5 Dans le contrôle *MenuStrip*, cliquez sur le menu **Fichier** pour faire apparaître les sous-menus et double-cliquez sur **Ouvrir**.



**Figure 7.8 :** *Méthode d'ouverture*

Une fois dans l'éditeur, ajoutez le code suivant :

```
If OpenFileDialog1.ShowDialog =  
Windows.Forms.DialogResult.OK Then  
Dim sr As  
New System.IO.StreamReader(OpenFileDialog1.FileName)  
Titre.Text = sr.ReadLine()  
NbPers.Text = sr.ReadLine  
TempsPrep.Text = sr.ReadLine  
CorpsText.Text = sr.ReadToEnd  
sr.Close()  
End If
```

Comme dans l'exemple précédent, vous utilisez le premier test de comparaison pour vérifier que l'utilisateur a bien cliqué sur le bouton OK de la boîte de dialogue. Ensuite, vous déclarez un nouveau flux de lecture, qui sera initialisé avec le contenu du fichier sélectionné dans la boîte de dialogue d'ouverture de fichier.

Vous lisez ensuite les trois premières lignes, qui vont respectivement contenir le titre, le nombre de personnes prévu et le temps de préparation. Une fois que ces trois lignes sont lues, il suffit de lire le fichier jusqu'à la fin pour avoir le corps de la recette.

Pour commencer la collection de recettes, voici celle du clafoutis aux cerises :



REMARQUE

### Le clafoutis aux cerises

Pour 6/8 personnes, temps de préparation : 10 minutes.

Ingrédients :

- 700 g de cerises bien mûres
- 2 œufs + 2 jaunes d'œufs
- 5 cuillerées à soupe de farine (100 g)
- 5 cuillerées à soupe de sucre roux (100 g)
- 1/4 de litre de lait (25 cl)
- 60 g de beurre
- 1 pincée de sel
- 1 cuillerée à café d'extrait de vanille
- 1 sachet de sucre vanillé (facultatif)

**Préparation :**

Lavez et équeutez les cerises.

Allumez votre four à thermostat 6/7 (180/200° C).

Dans un plat à tarte, mettez le beurre puis enfournez-le.

Battez les œufs en omelette, ajoutez la pincée de sel ainsi que le sucre.

Mélangez bien (au fouet ou au robot).

Ajoutez l'extrait de vanille, mélangez et jetez la farine en pluie.

Mélangez bien jusqu'à obtenir un liquide épais et onctueux.

Sortez le beurre fondu du four, ajoutez-le au liquide puis ajoutez le lait tout en remuant.

Une fois que vous avez mélangé, ajoutez les cerises à la préparation.

Avec un essuie-tout, étalez le beurre resté dans le plat pour le graisser, puis ajoutez la préparation.

Ajoutez quelques noisettes de beurre par-dessus.

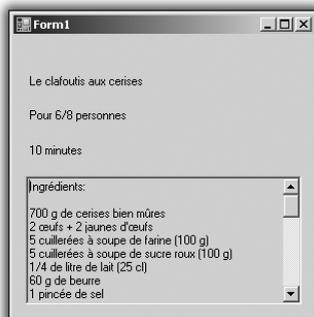
Enfourez 45 mn (jusqu'à ce que le dessus commence à bien dorer).

Dès la sortie du four, saupoudrez de sucre vanillé.

Servez tiède.

Utilisez maintenant le programme d'entrée de recettes pour sauvegarder cette recette de clafoutis dans un fichier, puis lancez le programme de lecture de recette pour récupérer le contenu du fichier.

En cliquant sur **Ouvrir** puis en sélectionnant le fichier de la recette de clafoutis, vous obtenez le résultat suivant :



**Figure 7.9 :**  
*Chargement de la recette*

Vous remarquez un petit problème ? La recette n'est pas du tout mise en page et le texte est difficilement lisible du fait qu'il n'y a aucun retour à la ligne. Pour remédier à cela, il suffit de remplacer le contrôle *TextBox* par une *RichTextBox*, offrant beaucoup plus de fonctionnalités.



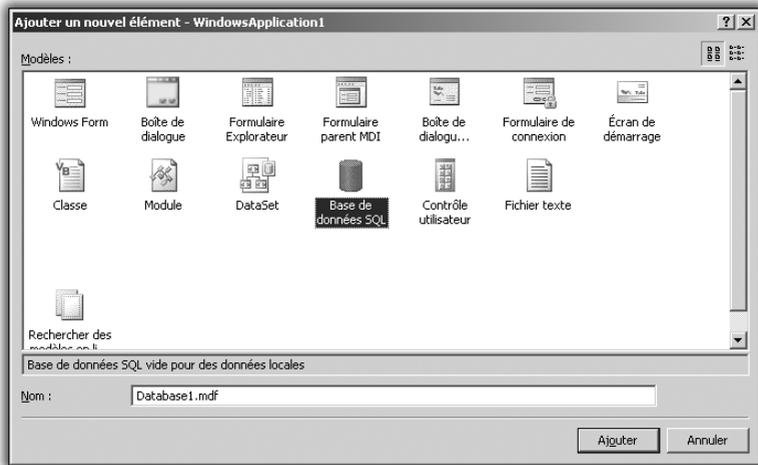
*Pour en savoir plus à ce sujet, lisez le chapitre Penser à l'utilisateur.*

## 7.5. Ajouter une base de données au projet

Maintenant que les fichiers n'ont plus de secrets pour vous, vous allez voir la deuxième méthode, consistant à sauvegarder des données dans une base. Les bases constituent un ensemble de fichiers organisés et regroupés en un seul bloc. Pour utiliser une base, il faut un moteur d'exploitation ou gestionnaire. Ici, vous allez utiliser SQL Server Express, qui, tout comme Visual Basic Express, est gratuit et disponible en téléchargement sur le site de Microsoft. Vous le trouverez à l'adresse <http://msdn.microsoft.com/vstudio/express/sql/>. Téléchargez le programme d'installation et lancez-le. Suivez les instructions comme lors de l'installation de Visual Basic Express. L'opération ne prend que quelques instants.

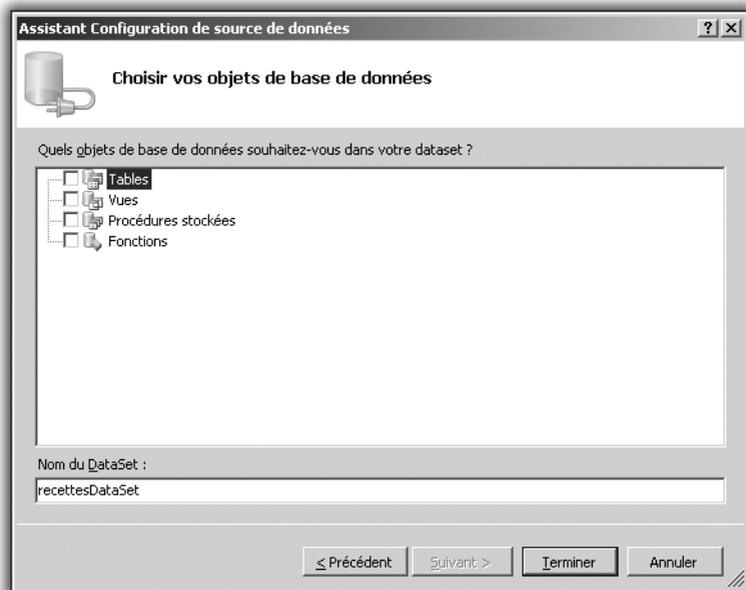
Une fois SQL Server Express installé, vous n'avez plus qu'à l'utiliser à travers Visual Basic Express. Vous allez maintenant reprendre le programme de gestion des recettes, en le rendant beaucoup plus performant par l'utilisation d'une base de données SQL Server Express.

- 1 Commencez par créer un nouveau projet de type application Windows dans Visual Basic Express.
- 2 Une fois dans le designer de formulaires, cliquez du bouton droit sur le nom du projet dans l'Explorateur de solutions puis cliquez sur **Ajouter un nouvel objet**.
- 3 Choisissez *Base de données SQL* (SQL Database).



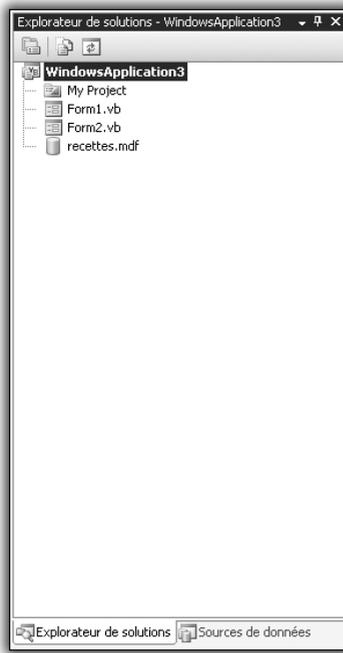
**Figure 7.10** : Sélection d'une base

- 4 Une fois la base sélectionnée, donnez-lui le nom `recettes.mdf` puis cliquez sur le bouton **Ajouter**. L'Assistant de configuration de source de données s'affiche.



**Figure 7.11** : Assistant de configuration de source de données

- 5 Ignorez cet Assistant et cliquez sur **Annuler**. L'Explorateur ressemble maintenant à la capture suivante, avec le formulaire principal et la base de données.



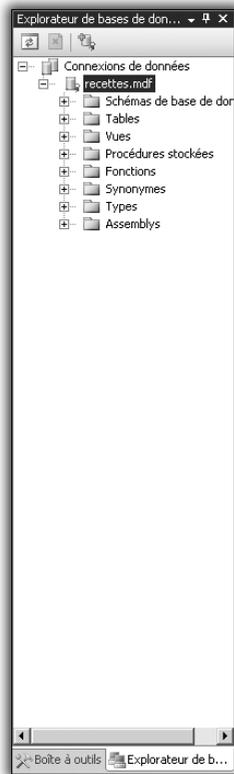
**Figure 7.12 :**  
*La base est ajoutée*

Une base de données est une sorte de classeur dans lequel vous pouvez à loisir incorporer des feuilles que l'on appelle "tables" et qui permettent d'organiser les données. Il faut maintenant réfléchir à la structure de ces tables et à la manière dont vous allez organiser les données. Le but ici est de ne pas répéter les données plusieurs fois dans le cas où vous auriez plusieurs tables. Le programme étant plutôt simple, vous avez besoin d'une seule table que vous allez représenter sous forme de tableau, comme décrit en introduction.

Un point important est que cette base de données est complètement indépendante de Visual Basic Express ou du langage Visual Basic, comme vous allez le voir maintenant.

Il faut à présent ajouter une table à la base. À gauche de l'interface de Visual Basic, un nouvel onglet est apparu, nommé **Explorateur de bases**.

- 6 Cliquez sur l'onglet **Explorateur de bases** pour développer la fenêtre afférente.



**Figure 7.13 :**  
*L'Explorateur de bases*

- 7 Cliquez du bouton droit sur le dossier **Tables** puis sélectionnez **Ajouter une nouvelle table**.

À la place du designer de formulaires s'affiche un tableau à trois colonnes, la première indiquant le nom de la colonne dans la table, la seconde le type de données que va contenir cette colonne et la troisième une boîte à cocher pour spécifier si oui ou non vous voulez que cette valeur puisse être nulle.

- 8 Vous allez vous conformer au schéma de table spécifié en introduction : une première colonne de la table représentera le nombre de personnes pour lequel la recette est prévue, une deuxième colonne le temps nécessaire, une troisième le nom de la recette, une quatrième la catégorie de plats et une cinquième le corps de la recette. Vous allez donc commencer par entrer les

spécifications de la première colonne, à savoir le nombre de personnes. Dans la case *Column Name*, entrez la valeur `NbPers`. Dans la colonne *DataType*, vous n'avez rien à entrer ; vous devez faire une sélection parmi une liste de types possibles. Pour symboliser un nombre entier simple, utilisez *numeric(18,0)*. Le tableau s'agrandit en ajoutant automatiquement une seconde colonne à remplir. Remplissez les lignes suivantes comme vous venez de le faire, en ajoutant une ligne `TempPrep` de type *ntext*, une ligne `Titre` de type *ntext*, une ligne `CatPlat` de type *Text*, et une ligne `CorpsRec` de type *text*. La différence entre les types *text* et *ntext* réside dans la longueur admissible des données que l'on peut entrer dans chaque case.

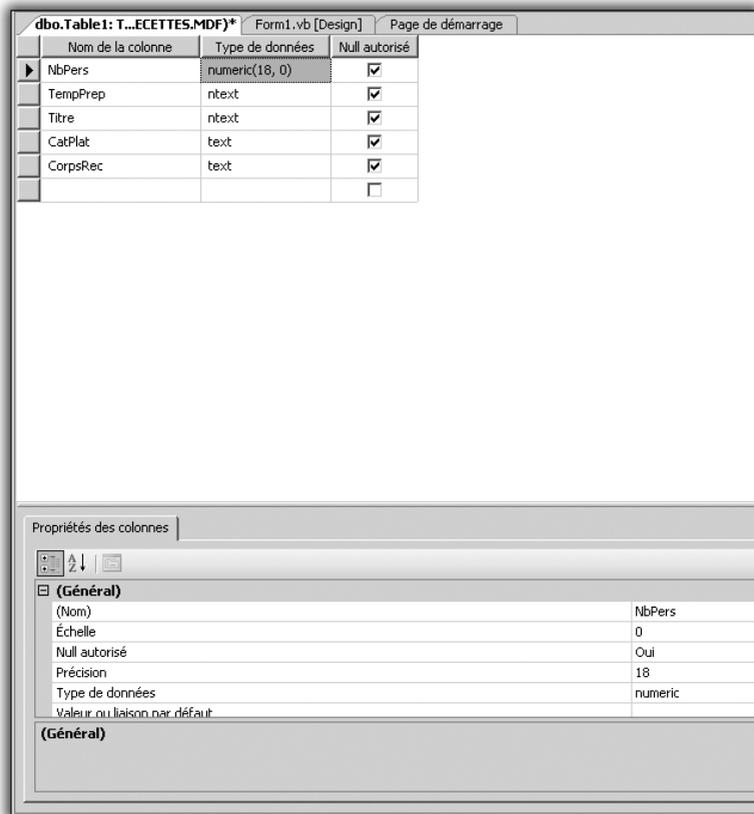


Figure 7.14 : Définition de la table

- 9 Sauvegardez maintenant la table que vous venez de créer en appuyant sur les touches **Ctrl+S**.

- 10** Un nom vous est alors demandé pour la sauvegarde de la table. Entrez `MaTable` puis cliquez sur le bouton OK.

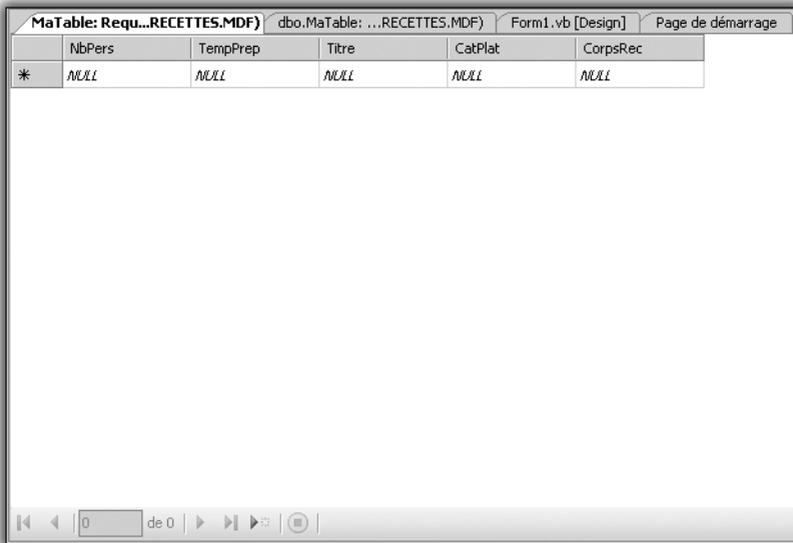
La table s'ajoute alors dans l'Explorateur de bases.



**Figure 7.15 :**  
*L'ajout de la table*

Vous allez maintenant ajouter des valeurs dans la table à partir de Visual Basic. Dans l'Explorateur de solutions, cliquez du bouton droit sur la table que vous venez d'ajouter puis sélectionnez **Afficher les données de la table**. Le designeur montre alors un tableau qui représente la table, les colonnes étant celles que vous venez d'entrer (voir Figure 7.16).

Ajoutez sur la première ligne les informations de la recette du clafoutis. Pour le nombre de personnes, faites attention à entrer un nombre entier pour respecter le format de type *numeric*. Vous êtes en effet moins libre que lors de l'utilisation des fichiers. Mais comme vous le verrez plus tard, cela permet de faciliter les recherches. Libre à vous d'ajouter d'autres lignes pour représenter d'autres recettes. Une fois que vous avez ajouté les recettes que vous voulez, passez à la section suivante.



	NbPers	TempPrep	Titre	CatPlat	CorpsRec
*	NULL	NULL	NULL	NULL	NULL

Figure 7.16 : Le designer de tables

## 7.6. Afficher les données de la base

Vous n'aurez jamais à saisir toutes les données à travers Visual Studio. Vous disposez d'outils pour afficher, modifier et rechercher les éléments dans la base. Visual Basic .NET permet en effet la manipulation des bases de données grâce à ADO .NET, qui peut être vu comme un ensemble de fonctions, méthodes et objets qui permettent d'effectuer toutes les opérations possibles sur les bases de données. Vous allez maintenant afficher les informations de la base de données dans le formulaire.

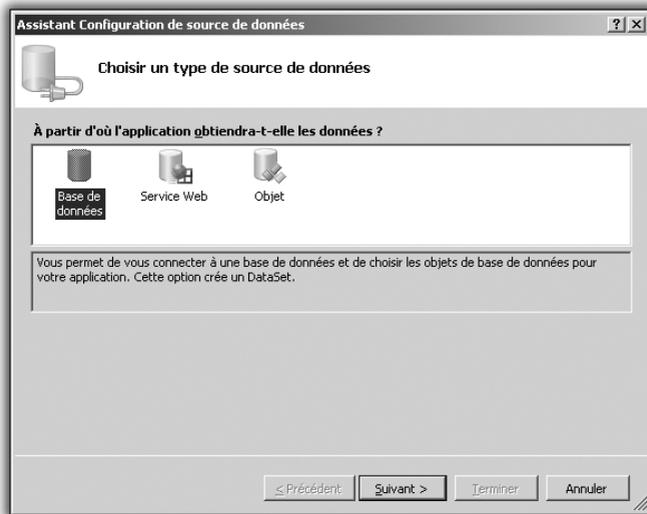
Revenez au designer de formulaires. Juste à côté de l'Explorateur de solutions, notez la présence de l'onglet **Sources de données**.

- 1 Cliquez sur l'onglet **Sources de données**, qui s'affiche à la place de l'Explorateur de solutions.



**Figure 7.17 :**  
*Les sources de données*

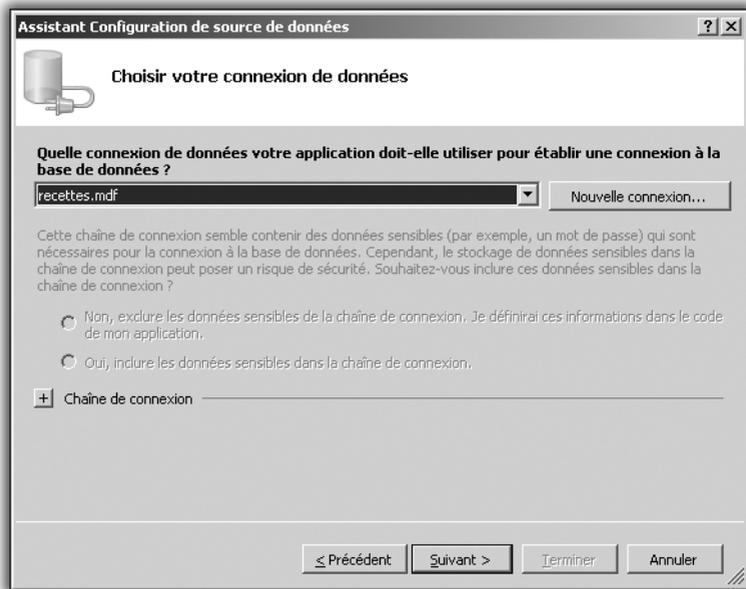
- 2 Cliquez sur le lien qui vous propose d'ajouter une source de données au projet. Choisissez *Base de données* et cliquez sur **Suivant**.



**Figure 7.18 :** *Le choix de la source*

L'écran suivant permet de choisir la base de données à utiliser comme source. Normalement, celle créée plus tôt et nommée *Recettes.mdf* est déjà sélectionnée.

**3** Cliquez sur **Suivant**.



**Figure 7.19** : Sélection de la base

- 4** Il vous est ensuite demandé si vous voulez créer ce qu'on appelle une chaîne de connexion. Répondez oui à cette question. Créer une chaîne de connexion permet de stocker quelque part dans le programme un lien qui pointe vers la source de données. Ainsi, si jamais la source de données change d'endroit, il suffira d'éditer le lien dans la configuration de l'application au lieu de reprendre toute la configuration de la base de données. Vous gagnerez ainsi un temps précieux. Répondez donc par l'affirmative puis cliquez sur **Suivant**.
- 5** Il vous est ensuite demandé quelles données de la source vous voulez reprendre. Vous voulez afficher vos recettes. Il faut donc reprendre toutes les informations de la table. Cliquez donc sur les boîtes de choix, comme indiqué sur l'écran suivant :

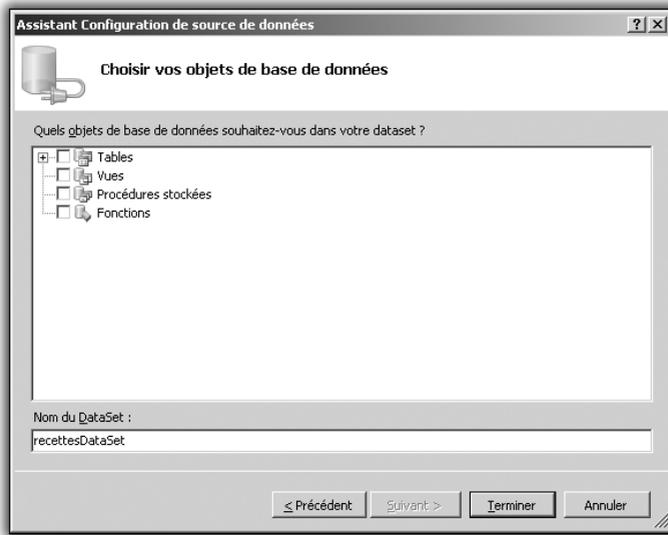


Figure 7.20 : Récupération de la table

- 6 Cliquez sur le bouton **Terminer**. La configuration de la source de données est complète.

La configuration complète de la source de données sert simplement à établir un lien entre la source de données, la base et le programme. Maintenant que vous avez un *DataSet* fonctionnel, il faut l'ajouter au formulaire. Retournez dans le designer de formulaires.



REMARQUE

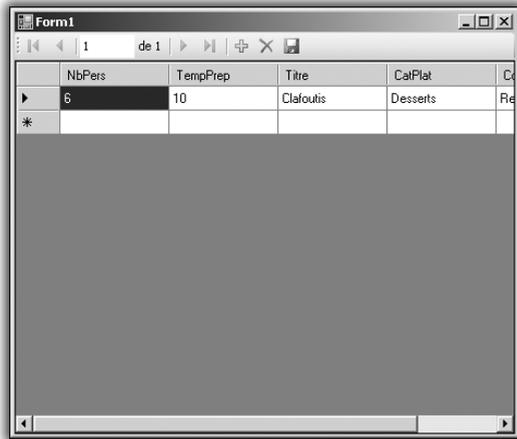
### Le DataSet

Le *DataSet* est un jeu de données qui fera pour vous le lien entre la base et les données recherchées. Vous pouvez l'éditer à tout moment dans la fenêtre des sources de données.

À partir de maintenant, dans la fenêtre qui montre les sources de données, lorsque vous cliquez sur le nom de la table, vous verrez apparaître sur la droite une icône de liste déroulante qui propose deux choix : *DataGridView* ou *Details*. Ces deux choix concernent la représentation des données de la base sur le formulaire. Pour un premier exemple, choisissez *DataGridView* et faites glisser le nom de la table vers le formulaire. Un tableau est ajouté au formulaire, ainsi que des contrôles de navigation et un contrôle *MaTableBindingSource*, qui est le

lien physique entre le formulaire et la table. Voyons maintenant comment utiliser tout cela.

Lancez une première fois l'application.



**Figure 7.21** : L'application

La ligne relative au clafoutis s'affiche bien en première ligne. Observez les boutons situés dans la barre d'outils en haut de la fenêtre d'application. Ils servent à naviguer parmi les enregistrements de la base de données. Cliquez sur le signe + situé parmi les outils. Une ligne est alors ajoutée au tableau, vous permettant d'insérer un enregistrement dans la base. Ajoutez une salade par exemple. Cliquez sur l'icône en forme de disquette à droite de la barre d'outils pour enregistrer les changements. Quittez l'application. De retour dans Visual Basic, appuyez une nouvelle fois sur la touche (F5) pour relancer le programme. Vous serez déçu en constatant que la modification n'a pas été prise en compte et que la base de données ne contient que la ligne relative au clafoutis.

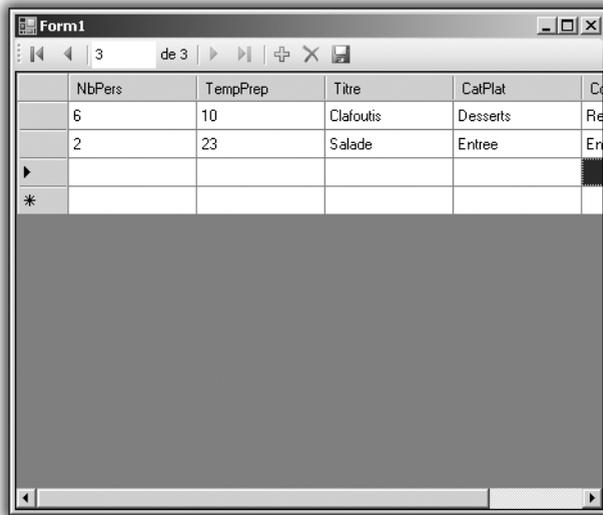
## Ne pas perdre la base

Lorsque que Visual Basic lance le projet, il génère la base de données et celle que vous aviez lors de votre précédente utilisation est écrasée par celle nouvellement générée, qui ne contient bien évidemment pas la modification. Rassurez-vous, il existe un moyen simple de remédier à cela.

- 1 Revenez dans Visual Basic, dans le designer de formulaires.
- 2 Cliquez sur la base de données dans l'Explorateur de solutions, de manière à afficher ses propriétés.
- 3 Dans la liste des propriétés, cherchez celle qui se nomme *Copie vers le répertoire de destination*. Par défaut, sa valeur est *Toujours*.
- 4 Changez cette valeur par *Ne pas copier*.

Cela fait, si vous lancez le programme, une erreur survient lors du lancement. L'initialisation de la base est impossible. En utilisant cette méthode, il faudra toujours penser à copier la base de données dans le répertoire d'exécution du programme. Normalement, la base est composée de deux fichiers, l'un portant l'extension *.mdf* et l'autre l'extension *.ldf*. Ils sont tous deux stockés à la racine du projet. Par défaut, ce répertoire est *Mes Documents\Visual Studio 2005\Projets\Nom du projet\Nom du projet*.

- 5 Copiez ces deux fichiers dans le répertoire *Bin\debug\* du répertoire de projet.
- 6 Si vous avez changé la propriété dans l'éditeur de propriétés de la base du designer de formulaires, vous avez fini. Relancez maintenant le programme et tentez une modification en ajoutant une valeur.



**Figure 7.22** : L'application après l'ajout d'une deuxième ligne

Fermez maintenant le programme, revenez sous Visual Basic.

- 7 Affichez les données de la table de recettes et regardez le nombre de lignes.

La ligne concernant la salade n'apparaît pas. Relancez le programme. La ligne apparaît bien dans le tableau de visualisation de la base. Étrange ? Pas du tout. Gardez bien à l'esprit que vous travaillez maintenant sur une copie de la base qui est située dans le répertoire de génération du programme. Les modifications ne sont donc pas synchronisées. Si vous modifiez la base dans Visual Basic, il faudra répercuter les nouveautés en copiant les informations de la base dans le répertoire encore une fois.

- 8 Retournez maintenant dans le designer de formulaires.

- 9 Nous avons énoncé plus haut deux manières d'afficher les données d'une base de données. Or vous n'en avez utilisé qu'une. Supprimez du formulaire le *DataGridView*, mais pas les outils de navigation. Dans la fenêtre des sources de données, cliquez sur le nom de la table, puis changez dans la liste déroulante la valeur de représentation en sélectionnant *Details*. La représentation se fera sous la forme d'un lot de contrôles, comme lorsque vous lisez à partir d'un fichier. En cliquant sur chacun des champs de la table, vous pouvez voir qu'ils disposent de la même petite icône de choix qui permet de sélectionner le type de contrôle qui les représentera. Choisissez *TextBox* pour chacun des contrôles pour le moment.

- 10 Dans le designer de formulaires, arrangez les contrôles comme bon vous semble, puis générez le projet en appuyant sur **F5**.

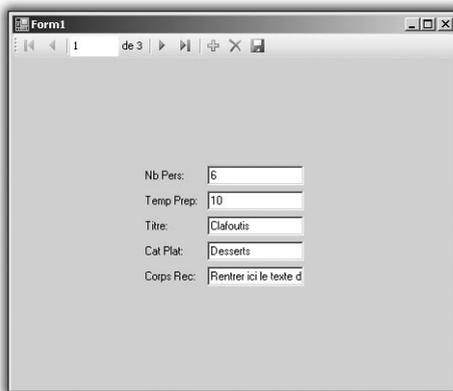


Figure 7.23 : La vue Details

Chaque champ est rempli avec la valeur associée dans la base. Utilisez les flèches droite et gauche situées dans la barre de menus pour naviguer dans les fiches de recette. Comme vous l'avez fait dans le tableau, vous pouvez ajouter une entrée en cliquant sur le bouton + de la barre d'outils et enregistrer les changements dans la base.

## 7.7. Aller plus loin grâce aux bases de données

Vous avez vu dans ce chapitre dédié à la sauvegarde de données qu'il était facile de sauvegarder les données d'un programme dans une base ou dans un jeu de fichiers manuellement définis. Les possibilités d'utilisation des bases de données dépassent le cadre de ce livre, mais sachez que vous pouvez avoir plusieurs tables dans une base et recouper les données entre elles. Dans l'application de gestion des recettes, cela permettrait par exemple de les classer, d'ajouter des fonctionnalités de recherche, en un mot d'aller plus loin dans la classification.

Gardez à l'esprit une ligne conductrice d'organisation. Le travail le plus long et le plus dur lors de la création d'une base de données est sa modélisation. Il en est de même pour l'organisation des données dans un système de fichiers. L'organisation de la base, pour être performante, nécessite un long moment de réflexion.

Efforcez-vous de reprendre cette application en améliorant la base de données et sa représentation dans le formulaire. Essayez également d'imaginer un moyen d'exporter les données de la base dans des fichiers pour assurer au programme une souplesse totale.