



Débuter avec Base



Version 1.0 du 30.04.2006

Réalisé avec : **OOo 2.0.2**

Plate-forme / Os : **Toutes**

Distribué par le projet fr.OpenOffice.org

Sommaire

| | |
|--|--------------------|
| 1 Présentation..... | 3 |
| 2 Quelques notions sur les bases de données..... | 3 |
| 3 Base intégrée HSQLDB..... | 3 |
| 4 Conception de la structure de la base..... | 3 |
| 5 Création du fichier Base..... | 4 |
| 6 Création de la table T_ADHERENTS..... | 5 |
| 6.1 Création de la clé primaire..... | 5 |
| 6.2 Création des autres champs..... | 6 |
| 7 Création de la table T_COMPTA..... | 7 |
| 8 Mise en place des relations entre tables..... | 8 |
| 8.1 Relation de T_ADHERENTS vers T_COMPTA..... | 8 |
| 8.2 Autre usage des relations..... | 10 |
| 9 Saisie de données sans formulaire..... | 11 |
| 10 Formulaires simples..... | 12 |
| 10.1 Création d'un formulaire des saisie avec l'assistant..... | 12 |
| 10.2 Création d'une zone de liste..... | 15 |
| 11 Requêtes simples..... | 19 |
| 12 Pour aller plus loin..... | 22 |
| 13 Crédits | 23 |
| 14 Licence..... | 23 |

1 Présentation

Ce document est destiné aux utilisateurs d'OpenOffice.org 2 (OOo 2) désirant utiliser le module Base et n'ayant pas ou peu de connaissances sur le sujet. Il présente quelques notions élémentaires sur les systèmes de gestion de bases de données relationnelles (SGBDR) et explique pas à pas le fonctionnement de Base en se basant sur un exemple de création d'une base de données simple.

2 Quelques notions sur les bases de données

L'objectif d'une base de données est le stockage structuré de données et la mise à disposition d'outils permettant d'afficher les données de manière pertinente. Une base est constituée de tables. Chaque table est constituée de champs, qu'on appelle aussi parfois colonnes. On peut envisager la structure d'une table (la description des champs qu'elle contient) comme un modèle de fiche à remplir. D'une certaine manière, une table peut-être vue comme une feuille de classeur Calc : les champs comme colonnes et les enregistrements comme lignes. Mais il-y-a une différence fondamentale : dans une table, le type de contenu des colonnes est indiqué dans la structure de la table, et est le même pour toutes les lignes d'une colonne. Chaque fois que l'on saisit une nouvelle fiche, on crée un enregistrement. Un champ est un espace réservé aux données auquel on a imposé certaines règles :

- ◆ type de données : on doit indiquer quel type de données le champ est susceptible de contenir. Par exemple, nombres entiers, ou chaînes de caractères. Nous verrons plus loin les types de données disponibles dans une base intégrée.
- ◆ contraintes : on peut imposer certaines contraintes aux données contenues dans un champ. Par exemple, la contrainte d'unicité indique au système que ce champ ne peut pas contenir deux fois la même valeur. La contrainte non-vide indique au système que ce champ doit impérativement contenir une donnée.
- ◆ valeur par défaut : on peut spécifier quelle valeur sera inscrite par défaut dans le champ si l'utilisateur ne saisit pas de données.
- ◆ clé primaire : Les bases intégrées imposent la présence d'une clé primaire dans chaque table. Une clé primaire est un champ (ou une combinaison de champs) qui sera l'identifiant unique de chaque enregistrement. Dans Base, cette clé primaire peut être générée automatiquement.

3 Base intégrée HSQLDB

Le module Base peut-être utilisé pour gérer une base de données externe (type MySQL), mais dans ce document vous allez créer une base intégrée HSQLDB. Les documents Base intégrés contiennent différents types d'objets :

- ◆ la structure de la base de données et les données stockées.
- ◆ les formulaires créés pour saisir, consulter ou modifier les données.
- ◆ les requêtes créées pour sélectionner les données à afficher.

Les bases intégrées présentent l'avantage que l'échange des données et des formulaires est simplifié puisque l'ensemble de ces éléments est contenu dans un seul fichier de type .odb que n'importe quel utilisateur d'OOo 2 pourra exploiter.

Par contre, les bases intégrées ne sont pas adaptées à une consultation simultanée par plusieurs utilisateurs, la gestion des droits des utilisateurs n'est pas facilement mise en place, et les bases intégrées HSQLDB présentent de faiblesses sur le plan des performances qui les rendent peu adaptées à une utilisation pour une base de grande taille.

Note importante : ce document est conçu autour de la création d'une base intégrée HSQLDB pour des raisons de simplicité (il n'y a pas besoin d'installer une autre application qu'OOo pour suivre la totalité du tutoriel). En conséquence, certaines fonctions décrites ici peuvent ne pas être

supportées par d'autres SGBDR.

4 Conception de la structure de la base

Avant de commencer à utiliser les outils fournis par Base, il convient de réfléchir un peu à l'organisation des données dans notre projet. Pour ce tutoriel, nous allons imaginer que nous souhaitons réaliser une base de données pour une association qui comprendra :

- ◆ un fichier des adhérents.
- ◆ un fichier des entrées/sorties bancaires afin de gérer les cotisations, subventions et dépenses.

Le fichier des adhérents devra contenir les informations d'identification d'un adhérent, ses coordonnées postales afin que les données puissent être utilisées pour des publipostages, et la date de prochaine échéance de cotisation de chaque adhérent afin de déterminer quels adhérents sont à jour de cotisation.

Le fichier des entrées/sorties devra permettre de saisir les opérations bancaires, d'attribuer un type à chaque opération (cotisation, remboursement de frais, etc), d'associer éventuellement un adhérent à une opération (lors d'une cotisation par exemple), et il devra comporter un indicateur permettant pour chaque opération de valider la prise en compte de l'opération par la banque. Cet indicateur sera utile pour connaître à tout moment le solde théorique d'un compte ainsi que le solde réel en fonction des opérations non enregistrées par la banque (un chèque déposé mais non encore crédité au compte de l'association par exemple).

Notre base comportera donc au moins deux tables : une table des adhérents et une table des opérations bancaires. Lors de la réalisation de la base, nous créerons en fait d'autres tables, les explications vous seront fournies à ce moment là.

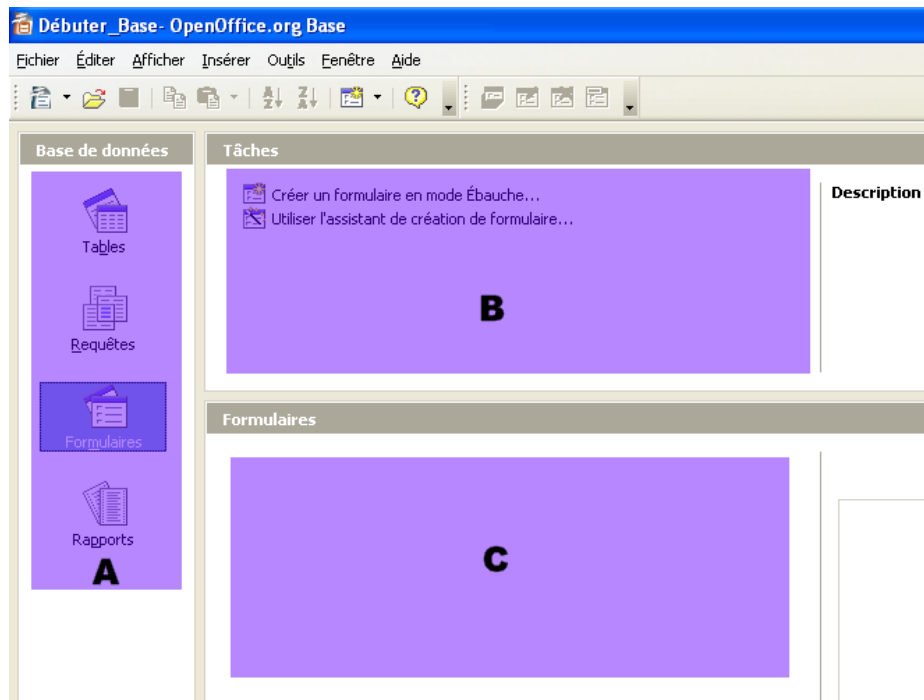
5 Création du fichier Base

Pour créer une nouvelle base, utilisez le menu Fichier > Nouveau > Base de données. Par défaut, l'option « Créer une base de données » est sélectionnée. Cliquez sur « suivant ». L'assistant vous propose alors deux paramètres à configurer :

- ◆ enregistrer la base dans OpenOffice.org. Si vous enregistrez la base dans OpenOffice.org, elle sera accessible en tant que source de données par l'ensemble des documents, pour insérer des données dans un courrier par exemple. Il est toujours possible de modifier ces réglages a posteriori, vous pouvez donc laisser le choix par défaut.
- ◆ ouvrir la base pour édition (et éventuellement création de tables avec l'assistant tables). Nous allons effectivement éditer la base, mais nous n'utiliserons pas l'assistant. Vous pouvez donc laisser le choix par défaut.

Cliquez sur « Terminer », vous pouvez sauvegarder la nouvelle base dans un répertoire de votre choix sous un nom qui vous convient. Le fichier Base s'ouvre alors en mode édition.

La colonne « Base de données » (zone A) comporte quatre icônes cliquables qui permettent d'activer les différents modes de travail :



Fenêtre principale de Base

- ◆ Tables
- ◆ Requêtes
- ◆ Formulaires
- ◆ Rapports

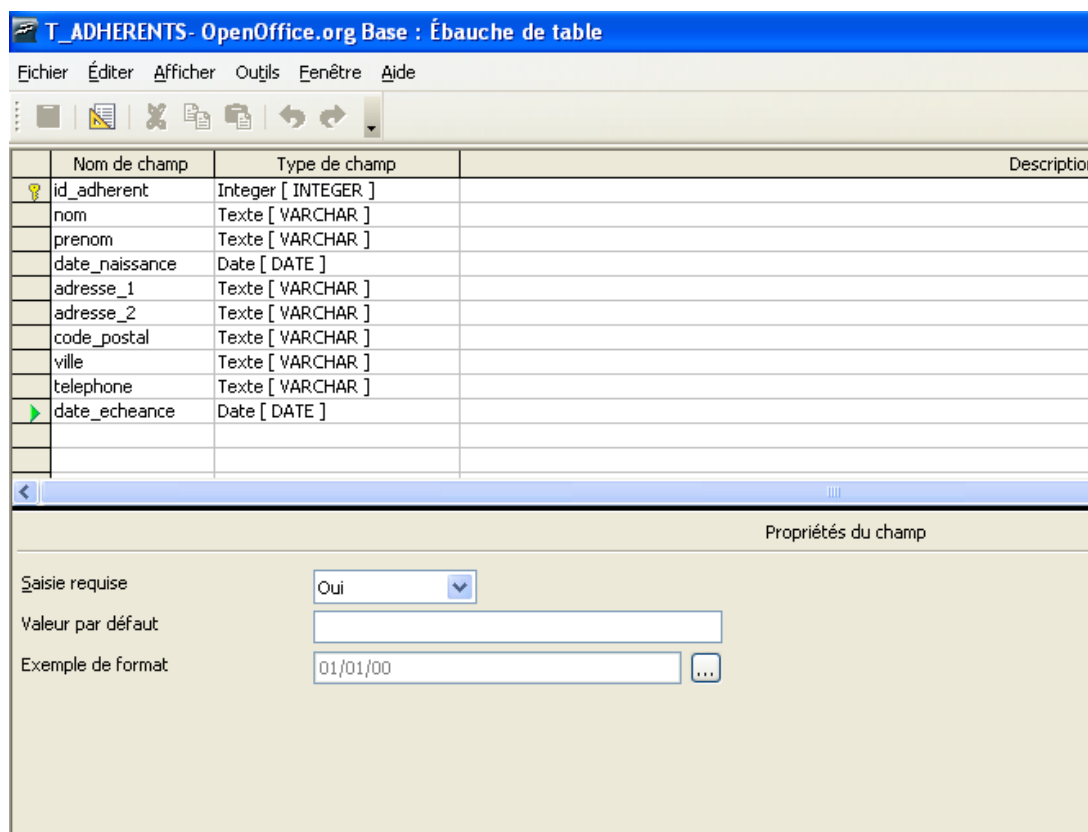
La zone « Tâches » (zone B) propose différents choix en fonction du mode de travail actif.

La zone C présente les différents objets créés dans chaque mode. Dans la figure ci-dessus, cette zone est vide puisqu'aucun formulaire n'a été créé pour le moment.

6 Création de la table T_ADHERENTS

En mode Table, cliquez sur « Créer une table en mode ébauche ». La fenêtre d'ébauche de table s'ouvre.

Comme évoqué plus haut, on voit que chaque champ est constitué par un nom de champ, un type de champ et des propriétés de champ. L'icône clé désigne la clé primaire d'une table. Le triangle vert indique quel champ est sélectionné pour l'affichage des propriétés.



Fenêtre ébauche de table après création des champs

Les bases intégrées HSQLDB supportent les caractères Unicode comme noms de tables ou de champs. Néanmoins, dans ce document, je n'utiliserai que des caractères non-accentués et je n'utiliserai pas d'espaces. Ceci permet de garantir un maximum de compatibilité dans le cas où la structure de la base doit être exportée vers un autre SGBDR. Les noms de tables seront en majuscules, précédés de « T_ ». Les noms de champs seront en minuscules, et les noms de clés primaires commenceront par « id_ ». C'est une habitude, mais cela n'a rien d'obligatoire pour une base intégrée HSQLDB.

6.1 Création de la clé primaire

Comme indiqué plus haut, Base impose la création d'une clé primaire pour chaque table. Cette clé primaire sera de toute façon nécessaire pour identifier de manière unique chaque adhérent, y-compris les homonymes. Pour créer notre clé primaire, il faut :

- ◆ attribuer un nom au champ. Nous l'appellerons « id_adherent ».
- ◆ définir le type de données du champ. Nous choisirons « Integer » (nombre entier).
- ◆ La zone « propriétés du champ » permet de modifier certaines propriétés du champ. Les choix proposés varient suivant le type de champ. Pour notre clé primaire, nous souhaitons que la valeur soit générée automatiquement par la base. Nous choisissons donc « auto-valeur : oui ». Le champ « expression d'auto-increment » bascule alors sur « Identity ». Cela signifie que le nombre généré sera bien unique (deux enregistrements de cette table ne pourront pas avoir le même nombre dans le champ « id_adherent »). Laissons les autres propriétés du champ à leur valeur par défaut.
- ◆ Pour indiquer que ce champ sera la clé primaire de cette table, il ne nous reste plus qu'à effectuer un clic-droit sur le triangle vert à gauche du nom du champ et sélectionner « clé primaire ».

Lorsque vous sauvegardez la table pour la première fois, vous devez lui attribuer un nom. Choisissez « T_ADHERENTS ».

6.2 Création des autres champs

Créez en suite un champ « nom », de type « varchar ». Le type varchar indique que vous voulez enregistrer une chaîne de caractères, et vous devez alors indiquer la taille maximale de cette chaîne dans la zone propriété de champ. Pour un nom de famille, une longueur maximum de 50 paraît raisonnable « saisie requise » permet d'obliger l'utilisateur à saisir un nom lors de la création d'un nouvel enregistrement. Cela paraît raisonnable de ne pas créer de fiche pour un adhérent dont on ne connaît pas le nom, vous pouvez donc sélectionner « oui ». En suivant les mêmes règles, créez un champ « prenom » avec les mêmes propriétés.

Créez ensuite un champ « date_naissance » de type « Date ». Ce type est destiné à contenir des informations de date. Les informations nom, prénom et date de naissance étant le minimum pour identifier civilement un individu, vous devriez également sélectionner la propriété « saisie requise : oui ». Vous pouvez également choisir un format dans lequel les dates seront affichées lors de la lecture de la table. Pour le moment, vous pouvez laisser le choix par défaut (« 01/01/00 »), vous pourrez modifier cette présentation ultérieurement en fonction des besoins de présentation.

Vous allez ensuite séparer les informations d'adresse postale en plusieurs champs. Créez un champ « adresse_1 », de type « varchar », propriétés « longueur = 50 », « saisie requise : non ». En effet, vous pouvez être amené à créer une fiche adhérent sans connaître forcément son adresse, qu'il sera toujours possible de saisir ultérieurement.

Un clic-droit sur le triangle vert permet de copier la définition du champ et de la coller sur la ligne du dessous. Vous n'avez plus qu'à modifier le nom du champ dupliqué en « adresse_2 ». Il est utile d'avoir deux champs afin de pouvoir séparer les informations telles que « bâtiment A » ou « escalier C ». Dans beaucoup de cas, le champ « adresse_2 » restera vide.

Créez maintenant un champ « code_postal ». On pourrait être tenté de lui attribuer le type « Integer » puisqu'un code postal est composé de chiffres. Mais en fait, il s'agit bien d'une chaîne de caractères, même si elle n'est constituée que de chiffres. Vous n'effectuerez pas d'opérations mathématiques sur le code postal. Vous devez donc lui attribuer le type « varchar ». Par contre, un code postal est composé de 5 chiffres, vous pouvez donc attribuer la propriété « longueur = 5 » au champ « code_postal ».

Enfin, créez un champ « ville » de type « varchar » et de longueur 50.

Créez un champ « telephone » de type « varchar » (pour la même raison que le code postal), et de longueur 10 (on considère dans toute cette partie adresse que les adhérents résident tous en France, dans le cas d'adresse à l'étranger, il faudrait évidemment créer d'autres champs).

Créez maintenant un champ « date_echeance », de type « Date », et attribuez lui la propriété « saisie requise : oui ».

Voici un résumé des différents champs de la table « Adhérents » et de leurs propriétés à cette étape :

| Nom du Champ | Type | Propriétés |
|---------------------|-------------|---|
| id_adherent | Integer | Auto-valeur : oui expression d'auto-increment : Identity clé primaire |
| nom | Varchar | saisie requise : oui longueur : 50 |
| prenom | Varchar | saisie requise : oui longueur : 50 |
| date_naissance | Date | saisie requise : oui |
| adresse_1 | Varchar | saisie requise : non longueur : 50 |
| adresse_2 | Varchar | saisie requise : non longueur : 50 |
| code_postal | Varchar | saisie requise : non longueur : 5 |
| ville | Varchar | saisie requise : non longueur : 50 |
| telephone | Varchar | saisie requise : non longueur : 10 |
| date_echeance | Date | saisie requise : oui |

N'oubliez pas de sauvegarder votre ébauche de table sous le nom «T_ADHERENTS» si vous ne l'avez pas déjà fait.

Vous pouvez fermer l'ébauche de table. Vous constatez que la table apparaît bien dans la fenêtre Base si le mode Tables est activé.

Si vous faites un double-clic sur l'icône de la table «T_ADHERENTS», une fenêtre s'ouvre qui présente la table sous forme de tableau, et vous pouvez saisir directement des données dans les champs. Vous pouvez vérifier que les contraintes mises en place fonctionnent correctement (champ « nom » vide par exemple, ou « code_postal » de plus de cinq caractères) : Base refuse d'effectuer l'enregistrement. Nous verrons plus loin qu'il existe des moyens plus adaptés d'effectuer la saisie ou la consultation des données. Pour le moment, effacez les enregistrements que vous avez effectués (clic-droit > effacer des lignes) et fermez la fenêtre.

7 Création de la table T_COMPTA

Créez une nouvelle table en mode ébauche, et de la même manière que pour la table précédente, créez les champs suivants :

| Nom du Champ | Type | Propriétés |
|---------------------|-------------|---|
| id_operation | Integer | auto-valeur : oui expression d'auto-increment : Identity clé primaire |

| | | |
|-------------------|---------|--|
| date_operation | Date | saisie requise : oui |
| type_operation | Integer | saisie requise : oui |
| ref_adherent | Integer | saisie requise : non |
| credit | Decimal | saisie requise : non décimales : 2 valeur par défaut : 0 Exemple de format : 0,00 € |
| debit | Decimal | saisie requise : non décimales : 2 valeur par défaut : 0 Exemple de format : 0,00 € |
| remarque | Varchar | saisie requise : non longueur : 50 |
| validation_banque | Boolean | saisie requise : oui valeur par défaut : non |

Remarquez deux champs particuliers : « type_operation » et « ref_adherent » qui sont de type Integer. Ces deux champs sont destinés à contenir des références à des enregistrements d'autres tables par le biais de relations que nous allons établir dans la suite du tutoriel. Remarquez également le champ « validation_banque » de type Boolean. Ce type de champ permet de stocker une information binaire (qui ne peut prendre que deux valeurs : Oui ou Non). Cela est parfaitement adapté à notre problème de validation d'une opération.

8 Mise en place des relations entre tables

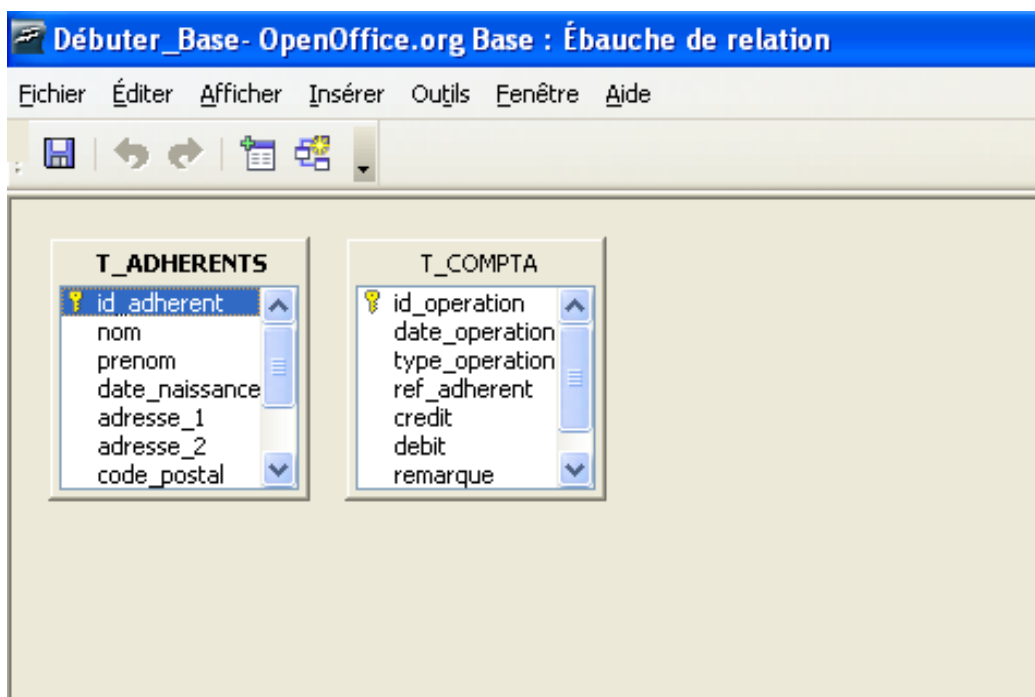
L'établissement de relations entre certains champs de différentes tables permet de lier logiquement certaines informations. Cela permet également, dans le cas d'une base intégrée, de déléguer au SGBDR le contrôle de la validité de certaines données.

8.1 Relation de T_ADHERENTS vers T_COMPTA

Puisque le champ « id_adherent » de la table T_ADHERENT identifie de manière unique un adhérent, nous allons l'utiliser pour indiquer dans le champ « ref_adherent » de la table T_COMPTA quel adhérent est concerné par une opération donnée. Lorsqu'aucun adhérent n'est concerné par une opération, le champ « ref_adherent » restera vide.

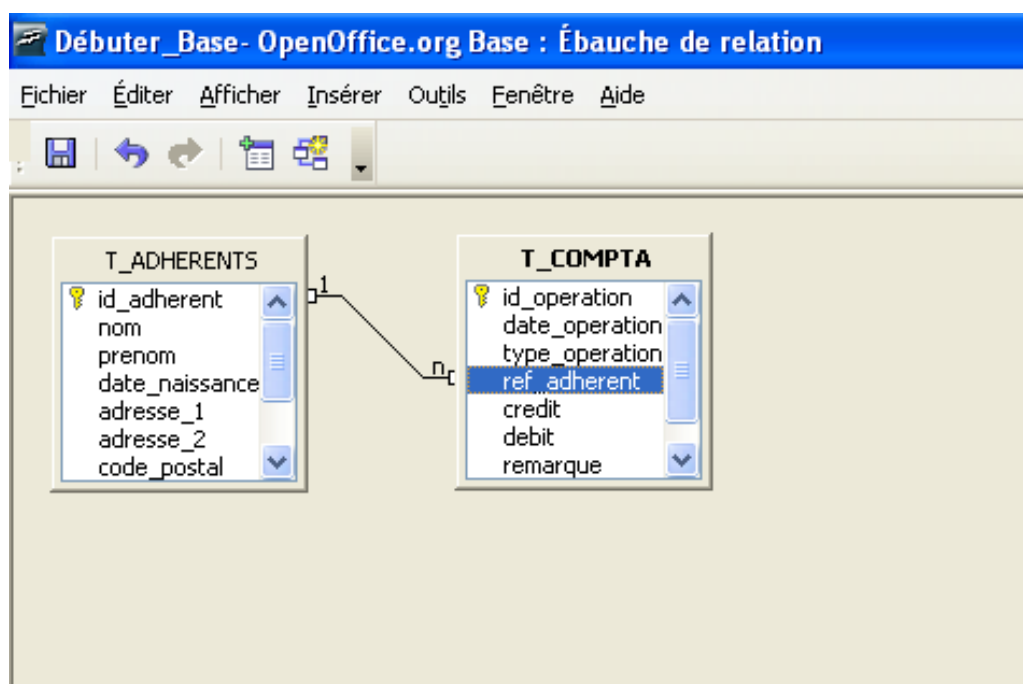
Pour établir une relation, sélectionnez le menu Outils > Relations ...

Cette fenêtre permet d'ajouter des tables concernées par une relation. Ajoutez donc T_ADHERENTS et T_COMPTA. Vous pouvez ensuite fermer cette fenêtre.



Fenêtre de gestion des relations (sans relation)

Pour établir la relation, cliquez sur le champ « id_adhérent » de la table T_ADHERENTS et tout en maintenant le bouton de souris pressé, glissez vers le champ « ref_adherent » de la table T_COMPTA. Relâchez ensuite le bouton de souris, une ligne apparaît entre les deux champs concernés.



Fenêtre de gestion des relations (relation établie)

Vous pouvez sauvegarder la relation et fermer la fenêtre.

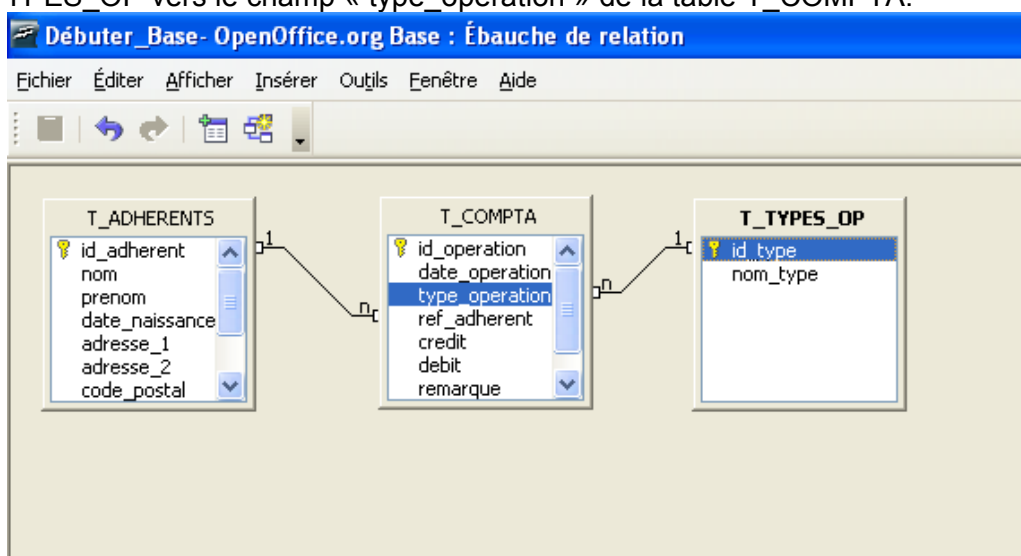
Dorénavant, vous ne pouvez saisir dans le champ « ref_adherent » que des valeurs qui existent dans le champ « id_adherent » d'un enregistrement de la table T_ADHERENTS. Si vous essayez de saisir dans « ref_adherent » une valeur qui n'existe pas comme « id_adherent », l'enregistrement sera refusé.

8.2 Autre usage des relations

Nous allons maintenant utiliser une relation pour limiter les choix de saisie dans le champ « type_operation » de T_COMPTA. Commencez par ouvrir une nouvelle ébauche de table et créez deux champs :

| Nom du Champ | Type | Propriétés |
|--------------|---------|---|
| id_type | Integer | Auto-valeur : oui expression d'auto-increment : Identity clé primaire |
| nom_type | Varchar | saisie requise : oui taille : 50 |

Enregistrez cette table sous le nom T_TYPES_OP. Cette table est destinée à contenir les noms des types d'opérations (cotisation, remboursement de frais, frais de fonctionnement, etc ...) que l'on souhaite associer à une opération bancaire. Lancez l'outil Relation, il devrait se rouvrir avec la précédente relation affichée. Sélectionnez le menu Insérer > Ajouter des tables et ajoutez la table T_TYPES_OP. Fermez le sélecteur de tables, la table T_TYPES_OP est maintenant présente dans la fenêtre de l'outil Relations. Créez une relation depuis le champ « id_type » de la table T_TYPES_OP vers le champ « type_operation » de la table T_COMPTA.



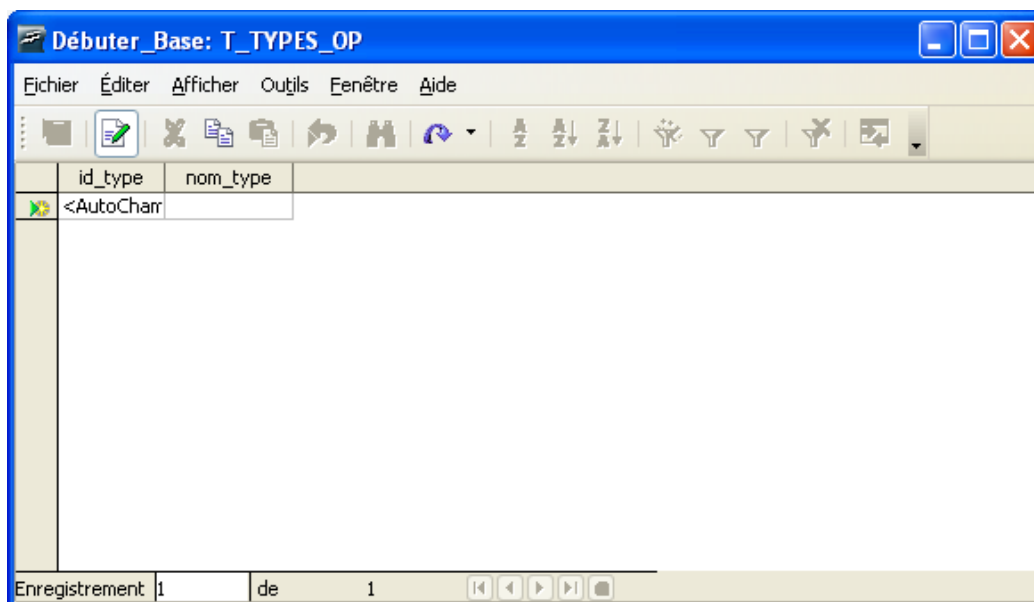
Fenêtre de gestion des relations (état final)

Sauvez les relations et fermez la fenêtre Relations.

Dorénavant, vous ne pouvez saisir dans le champ « type_operation » que des valeurs présentes dans le champ « id_type » d'un enregistrement de la table T_TYPES_OP. D'ailleurs, tant que vous n'avez pas saisi quelques types dans T_TYPES_OP, vous ne pouvez créer d'enregistrement dans T_COMPTA puisque le champ « type_operation » doit contenir une valeur, et que cette valeur doit être présente dans le champ « id_type » d'un enregistrement de T_TYPES_OP. Vous allez donc créer quelques enregistrements dans la table T_TYPES_OP.

9 Saisie de données sans formulaire

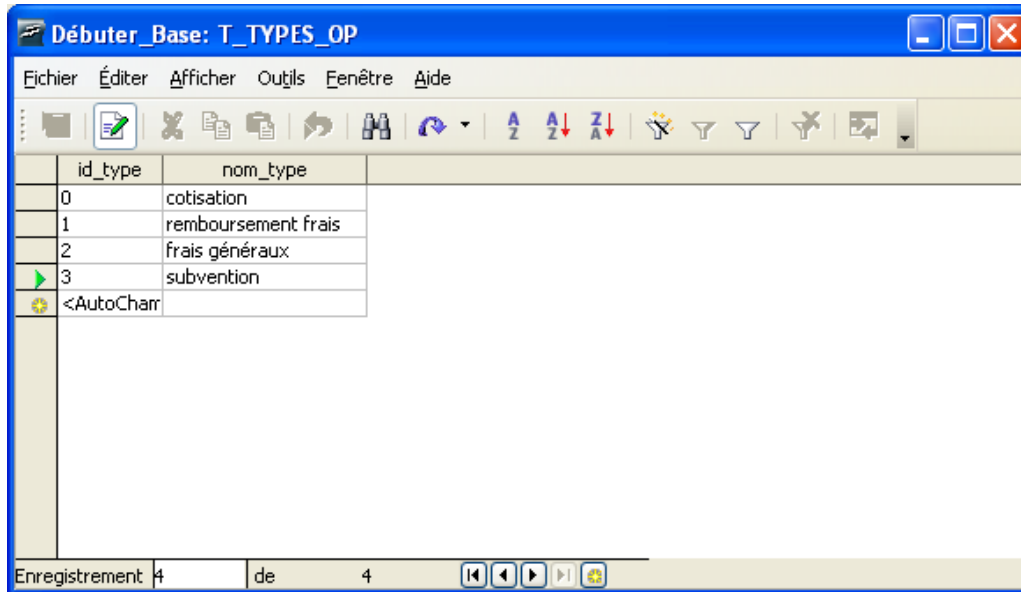
Depuis la fenêtre principale de Base, cliquez sur l'icône Tables dans la colonne Base de données. Les trois tables que vous avez créées apparaissent dans la zone Tables. Clic-droit sur la table T_TYPES_OP et sélectionnez « ouvrir ». La fenêtre de saisie s'ouvre.



Saisie directe en mode Table (aucun enregistrement)

La table s'affiche sous forme de tableau. Chaque colonne correspond à un champ, chaque ligne correspond à un enregistrement. La mention <AutoChamp> dans le champ « id_type » indique que ce champ se remplit automatiquement. Vous pouvez vous déplacer de champ en champ avec la touche TAB. Vous pouvez saisir quelques noms de type d'opérations. Les données saisies ne sont réellement inscrites dans la base que lorsque vous sauvegardez depuis cette fenêtre. Si vous fermez cette fenêtre sans avoir sauvegardé alors que vous avez rajouté ou modifié des données, un message d'alerte vous invitera à sauvegarder si vous voulez conserver les modifications. Vous devriez obtenir un écran de ce type :

Ce mode de saisie reste très basique et peu pratique pour des saisies de données complexes. Pour faciliter la saisie, vous aller utiliser des Formulaires.



Saisie directe en mode Table (quelques enregistrements créés)

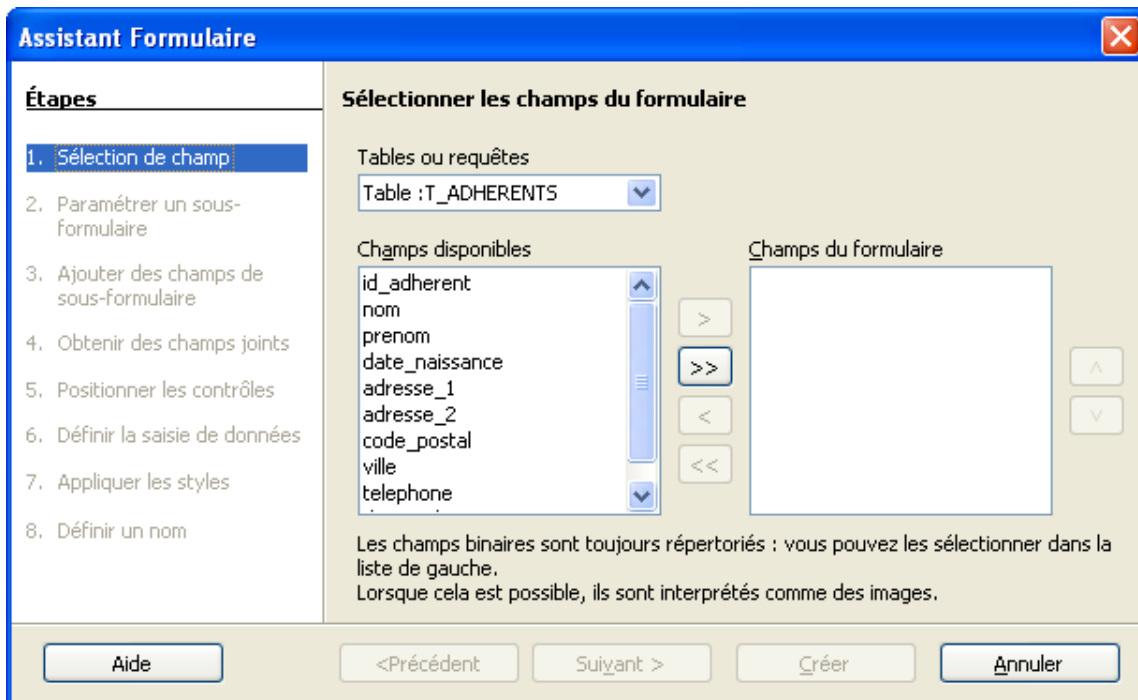
10 Formulaires simples

Les formulaires sont des documents Texte auxquels on a ajouté des propriétés de connexion avec une base, et qui permettent d'afficher, modifier ou saisir des données. Ils permettent de gérer la manière dont les données sont présentées.

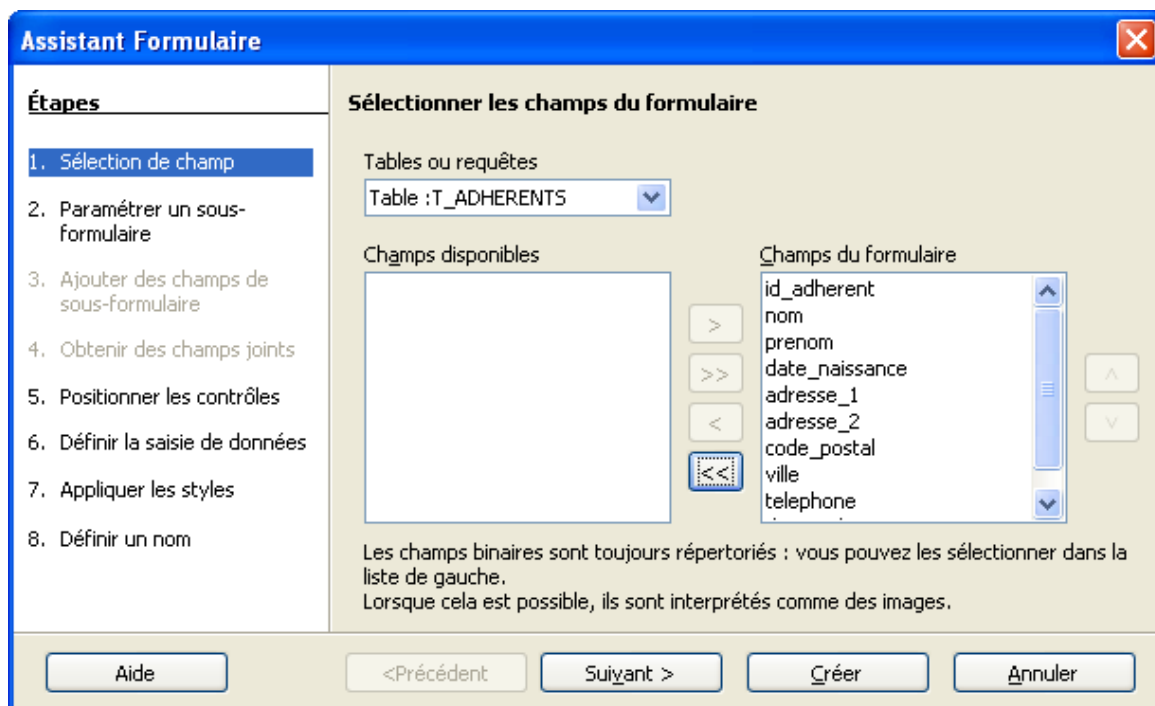
10.1 Création d'un formulaire de saisie avec l'assistant

Nous allons créer un formulaire de saisie pour la table T_ADHERENT avec l'aide de l'assistant de création de formulaires. Commencez par cliquer sur l'icône « Formulaire » dans la colonne « Base de données », puis cliquez sur la fonction « Utiliser l'assistant de création de formulaire » dans la zone « Tâches ». OOo crée un nouveau document texte, et le dialogue de l'assistant s'affiche par dessus.

Sélectionnez la table T_ADHERENTS avec la liste déroulante « Tables ou requêtes ». Les noms des champs de la table s'affichent dans la zone « Champs disponibles ». Puisque nous souhaitons travailler avec tous les champs de la table, cliquez sur le bouton >> pour transférer la totalité des champs disponibles dans la zone « Champs du formulaire ».



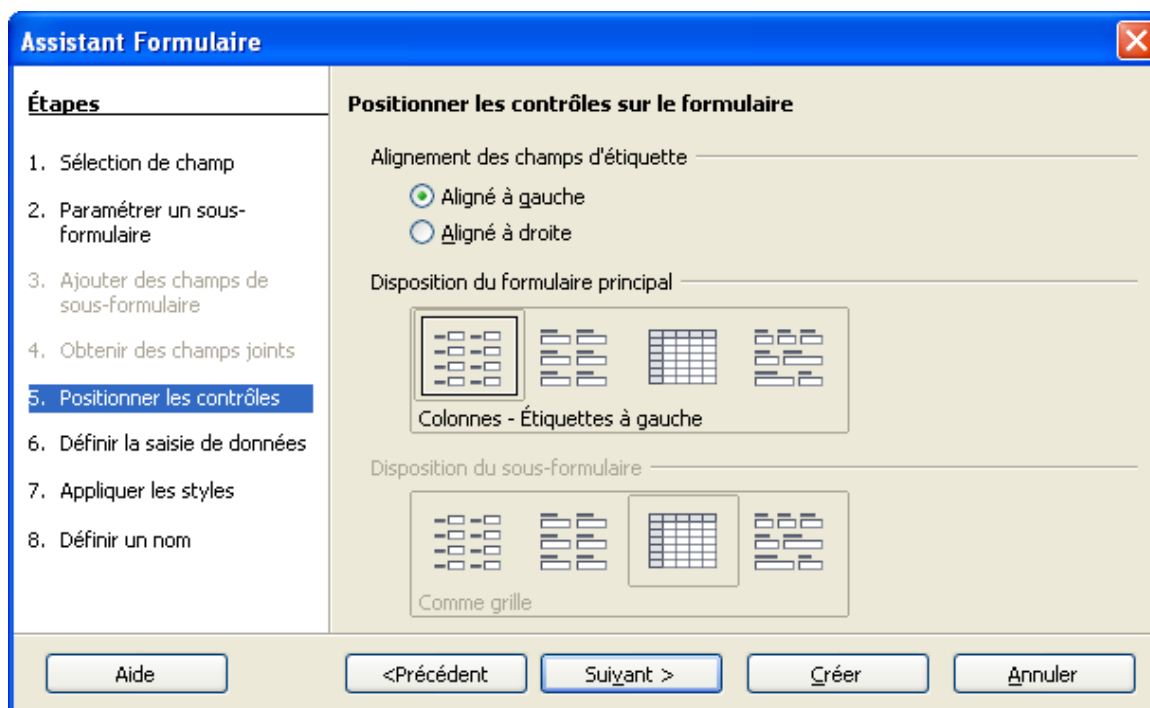
Première fenêtre de l'assistant de création de formulaires



Première fenêtre de l'assistant de création de formulaire après choix des champs du formulaire

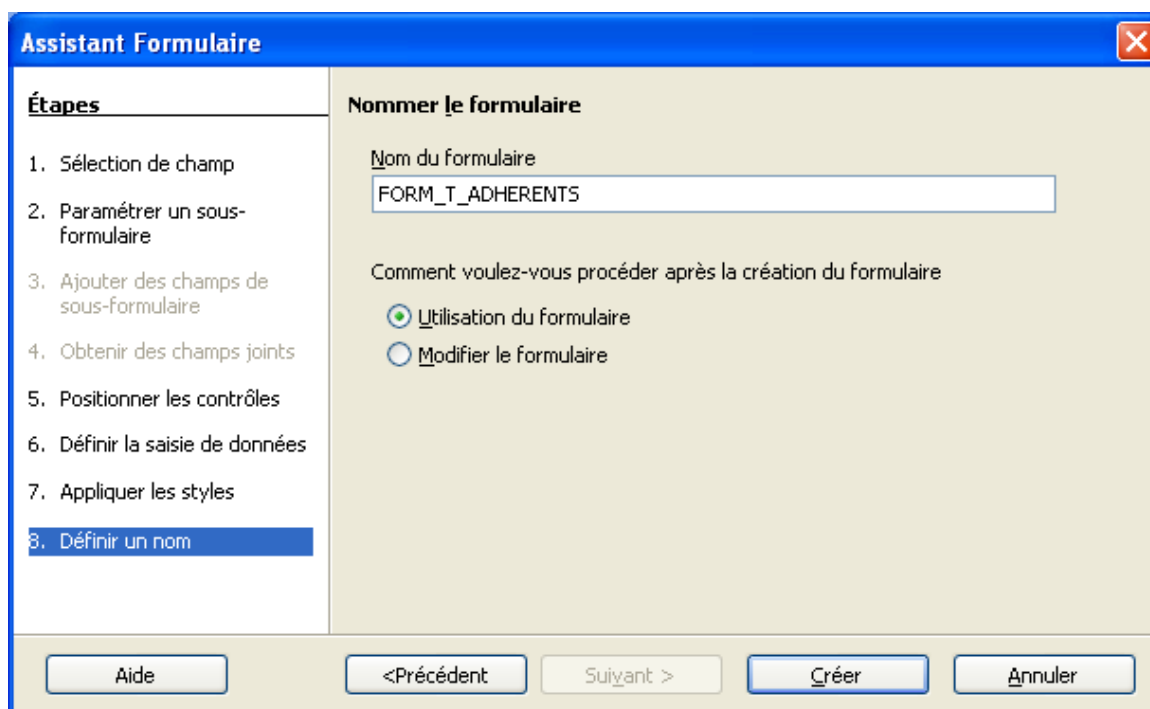
Nous n'utiliserons pas de sous-formulaire dans ce formulaire, vous pouvez donc passer directement à l'étape 5 en cliquant sur « Positionner les contrôles ».

Cet écran permet de choisir comment les données seront présentées dans le formulaire. Vous pouvez cliquer sur les différents icônes de la zone « Disposition du formulaire principal », le formulaire en arrière-plan réagit dynamiquement, vous permettant ainsi d'avoir une idée du rendu. Sélectionnez « Colonnes - Étiquettes à gauche ». Vous allez laisser les valeurs par



Fenêtre de positionnement des champs du formulaire

défaut dans les étapes 6 et 7, vous pouvez donc cliquer sur « Définir un nom » pour passer à l'étape 8.



Étape finale de l'assistant de création de formulaires

Par défaut, l'assistant vous propose le nom de la table comme nom du formulaire. Par souci de clarté, vous pouvez rajouter la mention FORM en début de nom (FORM_T_ADHERENTS par exemple). Nous n'allons pas éditer ce formulaire, vous pouvez donc laisser le choix « Comment voulez-vous procéder après la création du formulaire » sur « Utilisation du formulaire ». Il est toujours possible d'éditer un formulaire ultérieurement lorsque c'est nécessaire. Cliquez sur

« Créer », le formulaire est ouvert au premier plan.

The screenshot shows the OpenOffice Writer interface with a form titled "FORM_T_ADHERENTS(en lecture seule)". The form is displayed on a light orange background. The fields and their values are as follows:

| | |
|----------------|------------|
| id_adherent | <AutoChamp |
| nom | |
| prenom | |
| date_naissance | 17/04/06 |
| adresse_1 | |
| adresse_2 | |
| code_postal | |
| ville | |
| telephone | |
| date_echeance | 17/04/06 |

The status bar at the bottom indicates "Enregistrement 1 de 1".

Formulaire après création. La table ne contient pas de données

La mention « en lecture seule » dans la barre de titre signifie que vous n'êtes pas en mode d'édition de formulaire, mais vous pouvez tout à fait saisir des données dans votre table. Voici comment le formulaire se présente lorsque vous avez effectué quelques saisies :

La barre d'outils « Navigation pour formulaires » permet d'effectuer diverses opérations liées aux bases de données, notamment :

FORM_T_ADHERENTS(en lecture seule) - OpenOffice.org Writer

Fichier Éditer Afficher Insérer Format Tableau Outils Fenêtre Aide

id_adherent 1

nom dupond

prenom colette

date_naissance 23/03/58

adresse_1 Batiment C

adresse_2 33 rue de la galette

code_postal 24123

ville Balarie

telephone 0101010101

date_echeance 01/01/06

Enregistrement 1 de 2

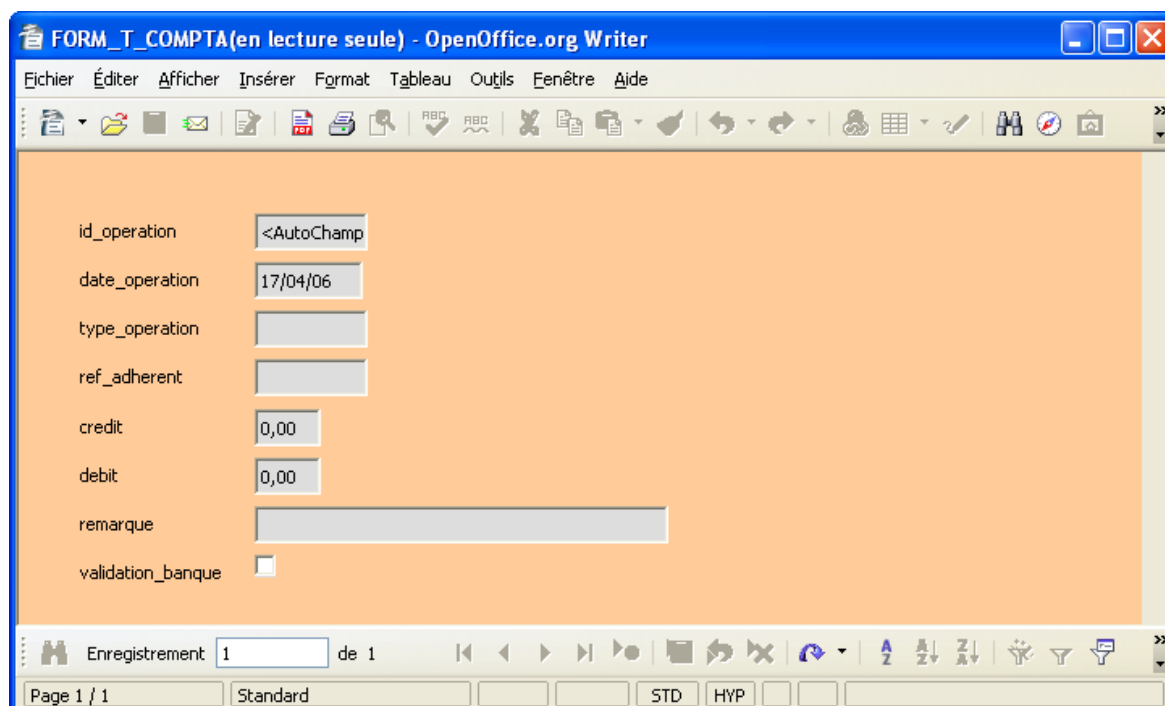
Page 1 / 1 Standard STD HYP

Formulaire avec quelques données saisies

- ◆ navigation parmi les enregistrements
- ◆ création d'un nouvel enregistrement
- ◆ validation (sauvegarde) d'un nouvel enregistrement ou d'une modification

10.2 Création d'une zone de liste

En utilisant la même méthode que ci-dessus, créez un formulaire pour la table T_COMPTA avec l'assistant. Vous obtenez cela :

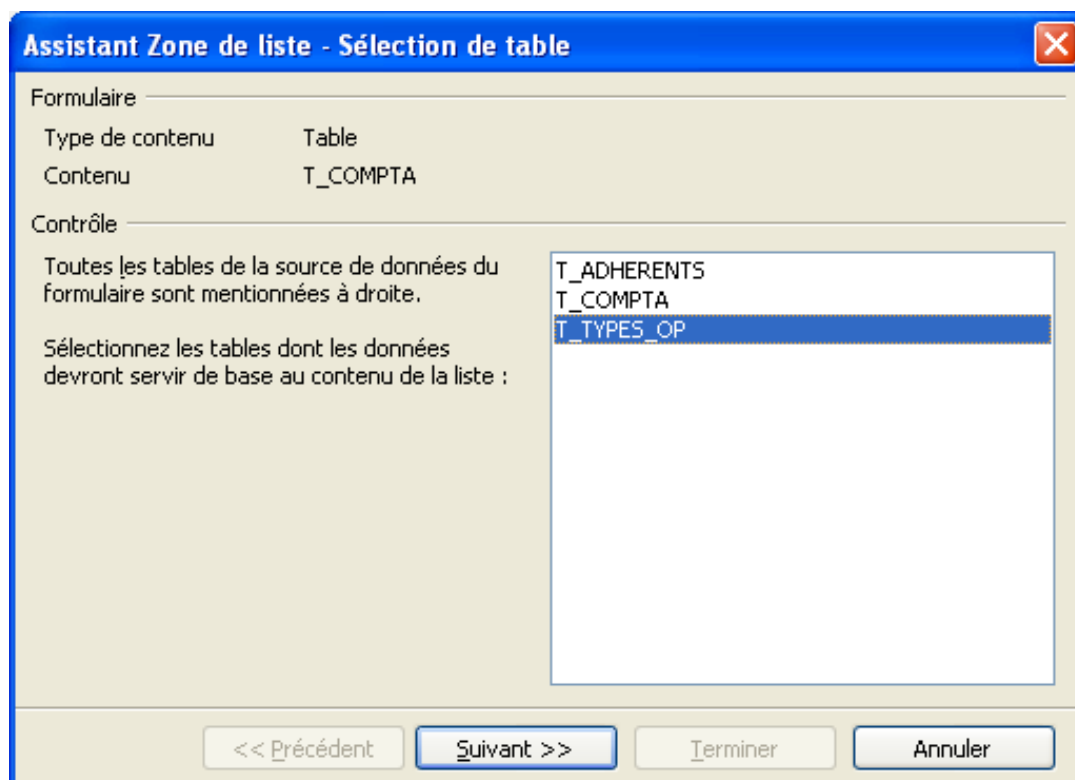


Formulaire FORM_T_COMPTA après création avec l'assistant

Comme dans l'exemple précédent, ce formulaire est fonctionnel immédiatement. Néanmoins, il pourrait être pratique que les deux champs « type_operation » et « ref_adherent » puissent être remplis par le biais d'une liste déroulante qui présenterait les choix autorisés pour ces champs. En effet, il n'est pas forcément pratique de saisir « 3 » dans le champ « ref_adherent » pour désigner l'adhérent dont « id_adherent » vaut « 3 ».

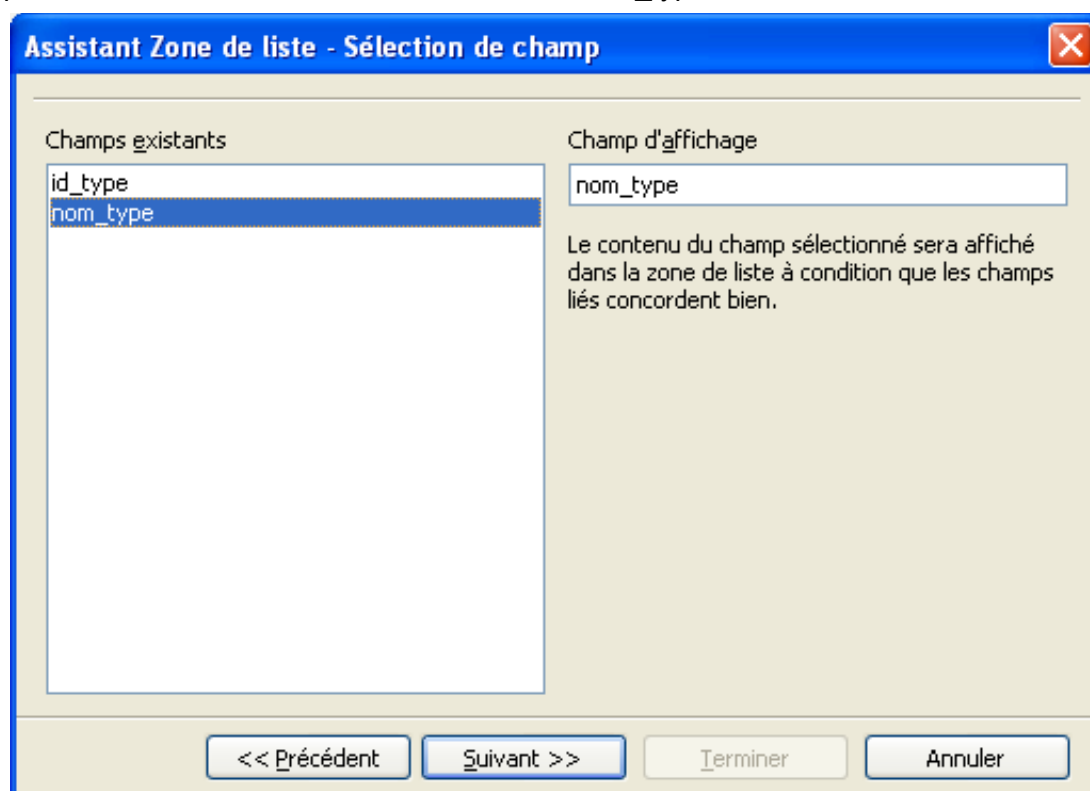
Pour éditer le formulaire, faites un clic-droit sur le nom du formulaire dans la zone « Formulaires » et sélectionnez « Modifier ». Le formulaire s'ouvre en mode édition. Vérifiez que les barres d'outils « Conception de formulaires » et « Contrôles de formulaires » sont bien activées. Si ce n'est pas le cas, menu Afficher > Barres d'outils pour les afficher.

Dans la barre d'outils « Contrôles de formulaires », cliquez sur l'outil « zone de liste » et dessinez ensuite un cadre à côté du champ « type_operation ». Un assistant zone de liste s'affiche qui va vous permettre de paramétrer la zone de liste. Le premier écran de l'assistant vous invite à choisir de quelle table les données à afficher doivent provenir. Dans notre cas, ce sera la table T_TYPES_OP.



Première fenêtre assistant zone de liste

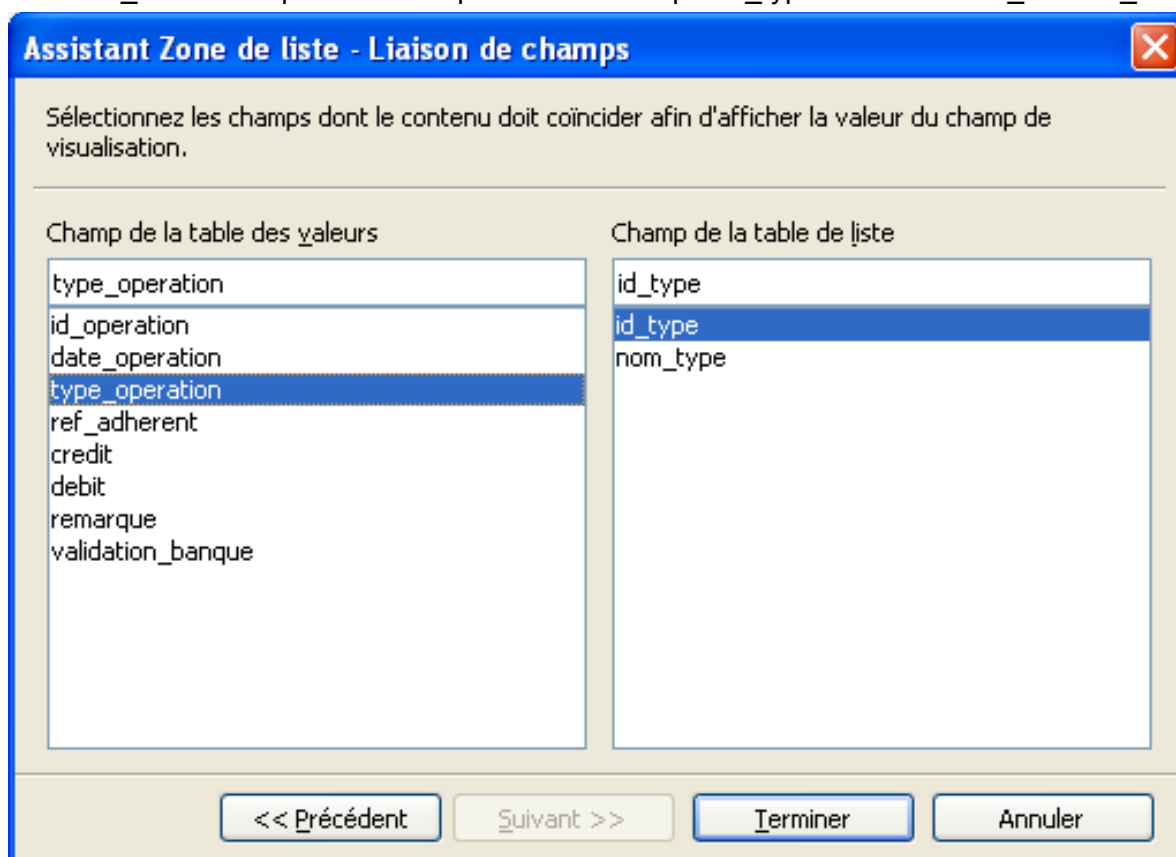
Cliquez sur « Suivant ». L'écran suivant permet de choisir quel champ de la table T_TYPES_OP doit apparaître dans la zone de liste. Choisissez « nom_type ».



Deuxième fenêtre assistant zone de liste

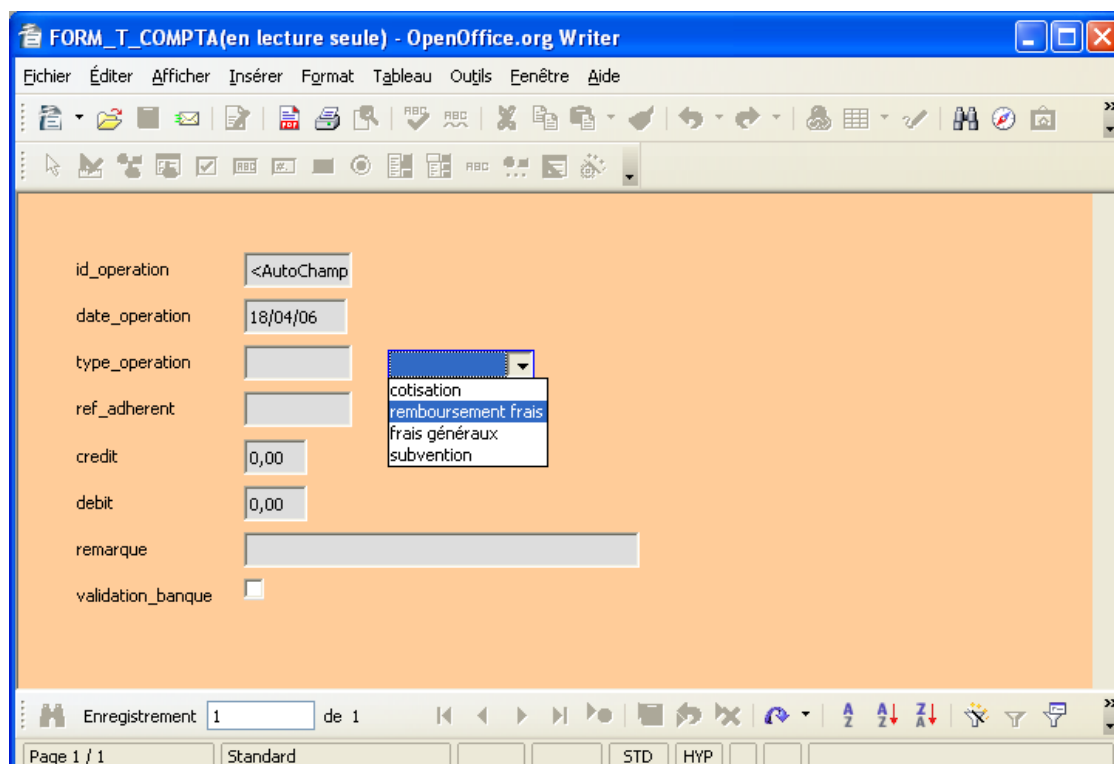
Cliquez sur « Suivant ». Le troisième écran permet d'indiquer quels sont les champs respectifs des deux tables concernées (T_COMPTA et T_TYPES_OP) qui doivent concorder pour que le

champ choisi dans la zone de liste s'affiche. Dans notre cas, c'est le champ « type_operation » de la table T_COMPTA qui doit correspondre au champ « id_type » de la table T_TYPES_OP.



Troisième fenêtre assistant zone de liste

Cliquez sur « Terminer ». Sauvegardez le formulaire modifié. Fermez la fenêtre d'édition de formulaire et rouvrez le formulaire en effectuant un double-clic sur son nom.



Formulaire avec zone de liste

Vous pouvez désormais remplir le champ « type_operation » en choisissant un nom dans la zone de liste.

De la même manière, vous pouvez créer une zone de liste pour saisir « ref_adherent ». Choisissez T_ADHERENTS comme table source, « nom » comme champ à afficher dans la liste, et choisissez « ref_adherent » et « id_adherent » comme champs liés.

Une étude complète des possibilités offertes par les formulaires demanderait beaucoup de place, nous allons nous arrêter là pour le moment.

The screenshot shows a window titled 'FORM_T_COMPTA - OpenOffice.org Writer'. The form is filled with the following data:

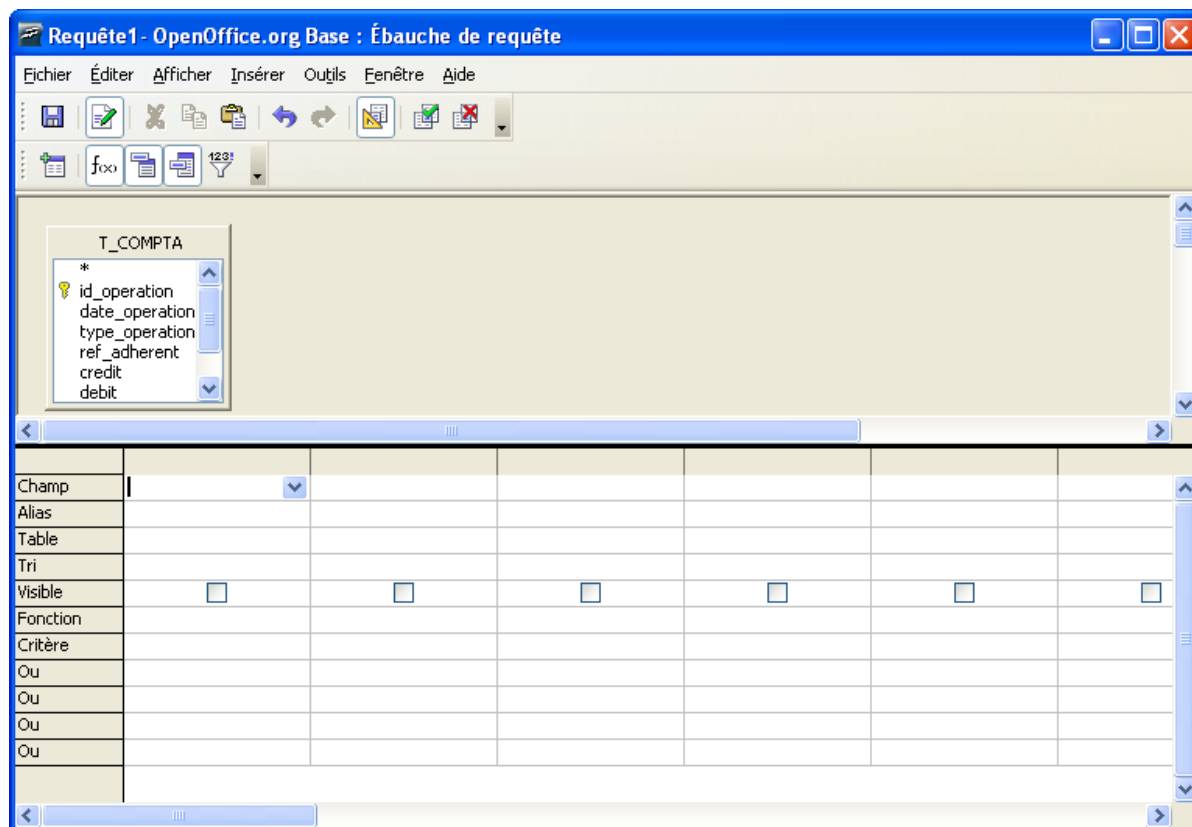
| | |
|-------------------|--------------------------|
| id_operation | <AutoChamp |
| date_operation | 18/04/06 |
| type_operation | 1 |
| ref_adherent | 0 |
| credit | 0,00 |
| debit | 43,50 |
| remarque | remboursement transports |
| validation_banque | <input type="checkbox"/> |

Formulaire terminé

11 Requêtes simples

Il serait utile de pouvoir connaître simplement l'état de notre compte bancaire, ainsi que l'état réel des nos finances (c'est à dire sans tenir compte de la validation par la banque). Une requête est une interrogation de la base de données selon des critères précis, et qui a donc l'avantage d'être toujours à jour puisque les nouveaux enregistrements sont pris en compte à chaque exécution de la requête. Nous allons donc commencer par créer une requête qui calcule la somme des crédits, la somme des débits et la différence entre les deux, ce qui nous donnera le solde financier courant.

Pour cela, passez en mode « Requêtes » et cliquez sur « Créer une requête en mode Ébauche ». Dans la fenêtre « Ajouter des tables », sélectionnez « T_COMPTA » et cliquez sur « Ajouter » puis « Fermer ».



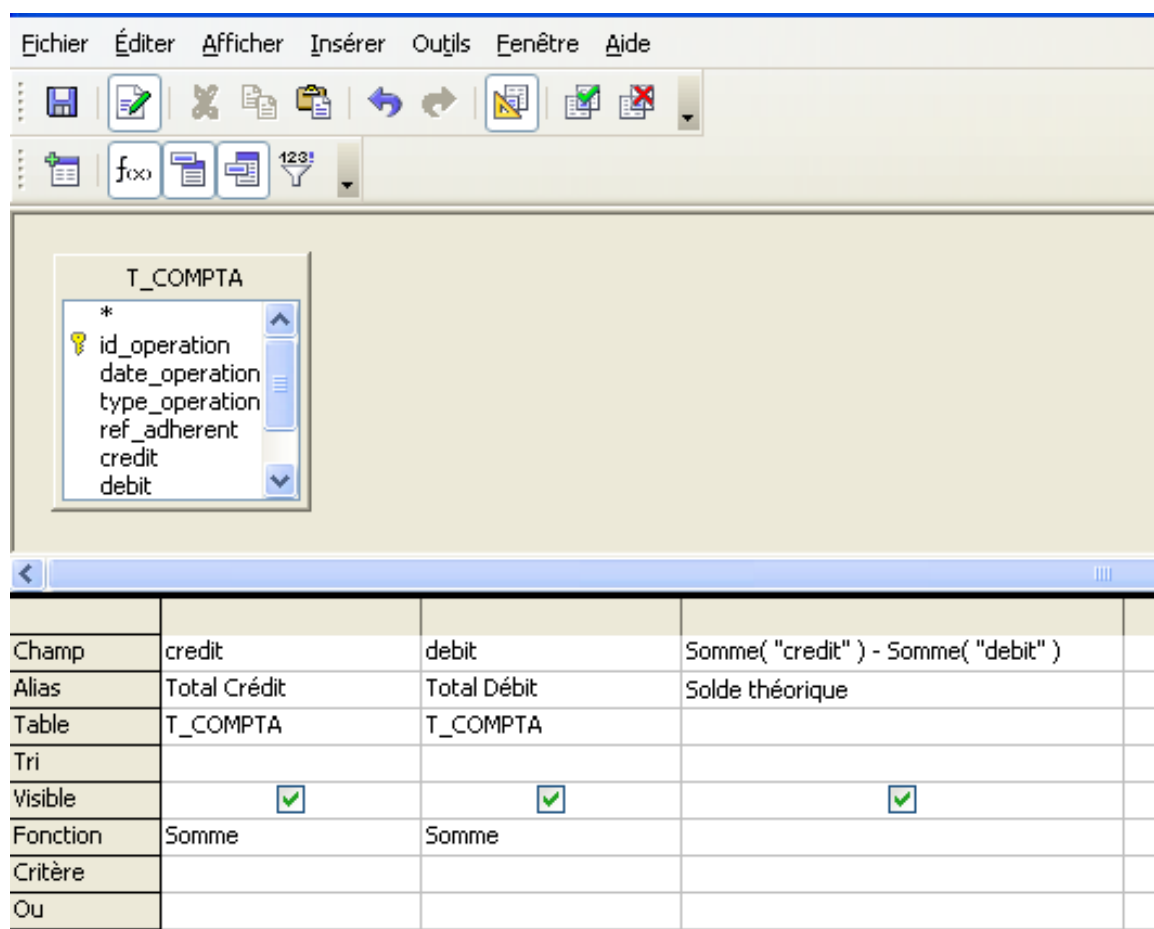
Fenêtre ébauche de requête

Cet outil va nous permettre d'indiquer quels champs nous souhaitons interroger, et éventuellement appliquer des fonctions.

Sélectionnez la première case Champ et choisissez « credit » dans le menu déroulant. Dans la ligne Alias (juste dessous) saisissez « Total Crédit ». Vérifiez que « visible » est coché, et dans la ligne « Fonction » choisissez « somme ».

Dans la deuxième colonne, choisissez « debit » comme champ et saisissez « Total Débit » comme Alias. Vérifiez que « visible » est coché, et dans la ligne « Fonction » choisissez « somme ».

Dans la troisième colonne, saisissez l'expression « Somme("credit") – Somme("debit") » dans la ligne « champ », saisissez « Solde Théorique » dans la ligne « Alias » et vérifiez que « visible » est coché.



| | | | |
|----------|-------------------------------------|-------------------------------------|--------------------------------------|
| Champ | credit | debit | Somme("credit") - Somme("debit") |
| Alias | Total Crédit | Total Débit | Solde théorique |
| Table | T_COMPTA | T_COMPTA | |
| Tri | | | |
| Visible | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Fonction | Somme | Somme | |
| Critère | | | |
| Ou | | | |

Fenêtre ébauche de requête complétée

Sauvez la requête. Fermez la fenêtre ébauche de requête. Vous pouvez désormais connaître l'état des finances de votre association en effectuant un double-clic sur la requête que vous venez de créer.

Pour connaître le solde en banque, nous allons utiliser le champ « validation_banque ». Rouvrez la requête précédente en mode édition en faisant un clic-droit sur le nom de la requête et en choisissant « modifier ». Lorsque la requête est ouverte en mode édition, enregistrez-la sous un autre nom. Vous allez simplement rajouter un critère sélection : ne faire la somme que pour les enregistrements dont le champ « validation_banque » vaut 1 (ou Vrai).

Dans la quatrième colonne de l'ébauche de requête, rajoutez le champ « validation_banque », décochez « visible », et dans la ligne « critère » saisissez "<> -1".

The screenshot shows the 'Requête2' window in OpenOffice Base. The design grid is as follows:

| Champ | credit | debit | Somme("credit") - Somme("debit") | validation_banque |
|----------|-------------------------------------|-------------------------------------|--------------------------------------|--------------------------|
| Alias | Total Crédit | Total Débit | Solde théorique | |
| Table | T_COMPPTA | T_COMPPTA | | T_COMPPTA |
| Tri | | | | |
| Visible | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Fonction | Somme | Somme | | |
| Critère | | | | <> - 1 |
| Ou | | | | |
| Ou | | | | |
| Ou | | | | |
| Ou | | | | |

Fenêtre ébauche de requête avec critère de sélection

L'exécution de cette requête vous donnera donc le solde réel en banque puisque ne seront prises en compte que les opérations validées.

12 Pour aller plus loin

Ce document présente une approche pratique de l'utilisation du module Base avec une base intégrée HSQLDB. Chacune des fonctions que nous avons brièvement examinées peut être utilisée pour exécuter des tâches bien plus complexes. Vous pouvez approfondir certains sujets en consultant les guides et how-to présents sur la [page de documentation](http://fr.openoffice.org) du site fr.openoffice.org.

Voici quelques liens utiles sur les bases de données en général et sur le langage SQL :

- ◆ [SQL](#)
- ◆ [Bases de données relationnelles](#)

13 Crédits

Auteur : **Manuel NAUDIN**

Remerciement : **Stéphane SANTON, Jean-François NIFENECKER, Tony GALMICHE, Guillaume LEBENE.**

Intégré par : **Tony GALMICHE**

Contacts : **Projet Documentation OpenOffice.org - fr.OpenOffice.org**

Traduction :

Historique des modifications :

| Version | Date | Commentaire |
|----------------|-------------|---|
| 0.1 | 18/04/06 | Première version |
| 0.2 | 27/04/06 | Modifications suggérées par Stéphane SANTON, Jean-François NIFENECKER, Tony GALMICHE. |
| 1.0 | 30/04/06 | Mise en ligne |
| | | |
| | | |

14 Licence

Appendix

Public Documentation License Notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the "License"); you may only use this Documentation if you comply with the terms of this License. A copy of the License is available at <http://www.openoffice.org/licenses/PDL.html>.

The Original Documentation is Débuter avec Base. The Initial Writer of the Original Documentation is Manuel Naudin Copyright © 2006. All Rights Reserved. (Initial Writer contact(s): audionuma@gmail.com).

Contributor(s): _____.
Portions created by _____ are Copyright © _____ [Insert year(s)]. All Rights Reserved.
(Contributor contact(s): _____ [Insert hyperlink/alias]).

NOTE: The text of this Appendix may differ slightly from the text of the notices in the files of the Original Documentation. You should use the text of this Appendix rather than the text found in the Original Documentation for Your Modifications.