

Tutoriel

Démarrer avec BASE

Les clés avec BASE de LibreOffice 3.3.1

Copyright

Démarrer avec BASE (ci-après le « Document ») est protégé par les droits d'auteur dont les contributeurs sont mentionnés dans la section ci-dessous. Vous pouvez modifier et distribuer le Document dans le respect de la licence Creative Commons CC BY-NC-SA (<http://creativecommons.org/licenses/by-nc-sa/3.0/>), version 3.0 ou supérieur, c'est à dire que tout travail dérivé du Document ne peut être utilisé à des fins commerciales, doit faire référence aux Contributeurs et doit être diffusé sous la même licence.

Tous les noms et les marques déposées mentionnées dans ce Document appartiennent aux détenteurs des droits respectifs.

Dans un esprit d'homogénéité avec la documentation existante sur *OpenOffice.org* et *LibreOffice*, la présentation du Document utilise le modèle *LibO3_3_chapter_template¹* (ci-après le « Modèle »). En aucun cas le présent Document n'est un travail dérivé du Modèle et, à ce titre, seul les droits et obligations de Copyright définie dans le présent Document sont applicables et ne recouvrent que le contenu du Document et non le Modèle dont il est issu.

Le Modèle de ce Document utilise des polices libres de droits que vous devrez télécharger et installer sur votre système le cas échéant et qui sont :

- Liberation Sans²
- Bitstream Vera Sans Mono³

Contributeurs

L'intégralité du contenu de ce document et des captures ont été réalisées par [Guillôme Copyright](mailto:guillome@cuk.ch) © 2011 – [Guillôme \(guillome@cuk.ch\)](mailto:guillome@cuk.ch)

Contact et Remerciements

Merci d'envoyer vos commentaires ou corrections à l'adresse suivante : guillome@cuk.ch

Date de publication et version du logiciel

Publié en 2011 sur Cuk.ch à l'adresse <http://www.cuk.ch/articles/4835>.

Basé sur LibreOffice 3.3.1

1 http://wiki.documentfoundation.org/cgi_img_auth.php/7/78/LibO3_3_chapter_template.ott

2 Téléchargeable à l'adresse suivante : <https://fedorahosted.org/liberation-fonts/>

3 Téléchargeable à l'adresse suivante : <http://ftp.gnome.org/pub/GNOME/sources/ttf-bitstream-vera/1.10/>

Contents

| | |
|---|----|
| Préambule..... | 4 |
| <i>Qu'est-ce que Base ?</i> | 4 |
| <i>Pourquoi Base ?</i> | 4 |
| <i>Les compétences préalables avant de se lancer dans BASE</i> | 5 |
| <i>Faisons une base de données : La gestion de nos dépenses</i> | 6 |
| <i>Référencement de la base de données</i> | 6 |
| <i>Création de la table principale</i> | 7 |
| <i>Import / Export</i> | 10 |
| <i>Formulaire de saisie</i> | 11 |
| <i>Liste déroulante</i> | 13 |
| <i>Et si on faisait des boutons ?</i> | 14 |
| <i>Macro : le B.A BA</i> | 15 |
| <i>Ouvrir un rapport / Fermer un rapport</i> | 15 |
| <i>Ouvrir un formulaire / Fermer un formulaire</i> | 15 |
| <i>Exécuter une requête SQL (Select, Update ou Insert)</i> | 16 |
| <i>Update et Insert en mode graphique</i> | 17 |
| <i>Deux macros spécifiques à notre projet</i> | 19 |
| <i>Assigner une Macro</i> | 21 |
| <i>Lancement d'une macro à l'ouverture d'un document BASE</i> | 22 |
| <i>Les Rapports</i> | 23 |
| <i>Conclusion</i> | 27 |

Préambule

Ce document regroupe un ensemble d'astuces ou d'informations pour créer une base de données avec BASE et des rapports associés. En effet, en tant qu'utilisateur novice de BASE, j'ai été assez dérouté par son interface et je pense que les informations regroupées dans ce document permettront à un débutant de faire ses premiers pas sans se décourager face à la première difficulté.

Ce document n'a pas pour but d'être un guide utilisateur pour BASE (pour cela, le lecteur se référera à des documents comme « Getting Started With Base »⁴)

Qu'est-ce que Base ?

N'étant pas un représentant de la Document Foundation, ni un affilié à Sun ou Microsoft, mon discours ici n'a rien d'officiel et il ne s'agit que d'un point de vue d'utilisateur basé sur ma propre expérience.

On présente souvent BASE comme une alternative à MS-ACCESS. Il n'en est rien.

En fait, en l'état, BASE est avant tout une application offrant la connectivité à de nombreux types de bases de données existantes et la possibilité d'opérer sur ces bases. BASE est donc davantage un outil de consultation/requêtes à destination des professionnels qu'un outil de création rapide de bases de données grand public.

BASE n'est pas une interface graphique pour créer aussi facilement des bases de données avec interfaces et rapports pour la gestion de sa cave ou de ses comptes qu'avec des produits propriétaires comme : Bento, Access... Néanmoins, cela ne veut pas dire qu'on ne peut pas le faire, et c'est justement l'objet de ce document de vous aider à le faire !

Pourquoi Base ?

Répondre à cette question est primordial avant de vous lancer dans votre projet ! Éliminons d'abord la mauvaise raison : la gratuité ! Si, pour vous, le seul critère est financier et que vous avez les moyens d'investir dans une licence propriétaire, ne vous embêtez pas avec BASE, vous serez découragé avant même de commencer (produit complexe, investissement communautaire nécessaire, performances décevantes, nombreux bugs existants...).

Concernant les performances, ma simple base de dépenses (quelques milliers d'enregistrement) provoque des délais en seconde sur les requêtes là où les traitements par FileMaker, que j'utilisai avant, prenaient un dixième de seconde. Les performances de HSQLDB via LibreOffice/Base sont catastrophiques!⁵

De bonnes raisons pour choisir BASE sont :

- un format ouvert et libre ;
- le support multiplate-forme ;
- la garantie de pérennité de votre solution ;
- un projet communautaire où vous pouvez être à la fois acteur et spectateur en bénéficiant de l'évolution du produit par la communauté ou en faisant évoluer le produit ;

⁴ http://wiki.documentfoundation.org/cgi_img_auth.php/8/8a/0108GS3-GettingStartedWithBase.pdf

⁵ Ce n'est pas moi qui le dit : http://fr.openoffice.org/Documentation/How-to/Bdd/09HSQLDB_OOo_0.5.pdf

Les compétences préalables avant de se lancer dans BASE

BASE, comme expliqué précédemment, n'est pas un outil graphique à portée du grand public. C'est une application professionnelle, puissante et complexe. Se lancer dans BASE nécessite, à minima, les compétences suivantes :

- La maîtrise du langage SQL ou, à défaut, une prédisposition intellectuelle à appréhender un langage informatique (vous pouvez vous référer au guide du langage SQL pour OpenOffice⁶);
- La capacité à faire de la programmation de type macro (ex : Visual Basic sous Excel), c'est à dire, avoir déjà programmé ou ne pas être effrayé par des lignes de commandes ;
- Connaître les concepts clés d'une base de données relationnelle
- Comprendre un minimum l'anglais (les ressources communautaires sont plus abondantes en anglais et le langage de macro est en anglais)

6 http://fr.openoffice.org/Documentation/How-to/Bdd/IntroSQL_Base.pdf

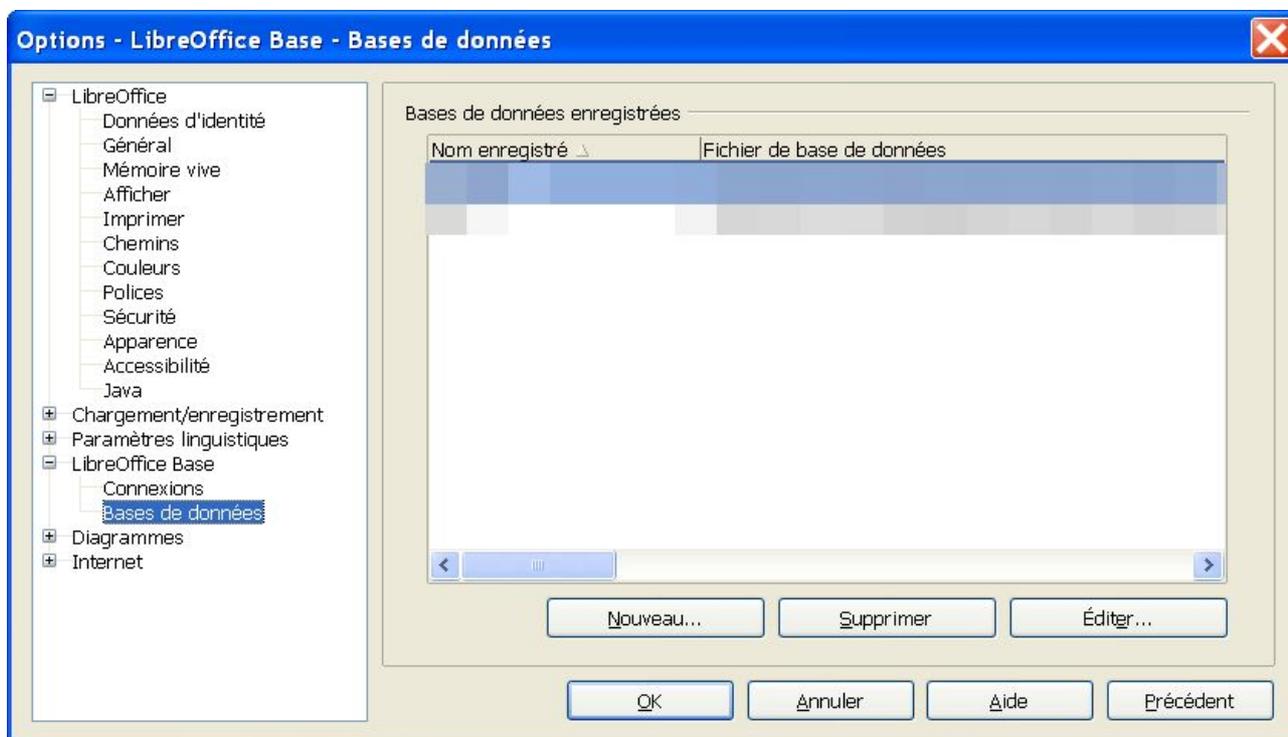
Faisons une base de données : La gestion de nos dépenses

Pour l'exemple, je vais reprendre une partie de la base de données que j'ai créée pour gérer mes dépenses. Ici, il ne s'agit pas de créer une application répondant à vos besoins ou de mettre en avant une méthodologie mais de donner des clés pour utiliser BASE afin de vous lancer rapidement et simplement dans un petit projet personnel.

Référencement de la base de données

Lors du lancement de BASE et de la création du fichier de base de données, l'assistant mentionnera pas défaut « le référencement de la base de données ». Ce référencement n'est pas anecdotique si vous êtes amené à déplacer votre base de données dans un autre environnement car il faudra configurer BASE.

Pour le référencement, sur PC, il faut aller dans le menu Outils → Options, puis choix LibreOffice Base → Base de données. Sur Mac, il faut aller dans le menu Base → Préférences.



Cliquez sur Nouveau pour ajouter le chemin de la base à référencer si besoin

A noter que les macros qui se basent sur le référencement de la base pour faire des appels simplifiés provoqueront une erreur d'exécution si la base n'est plus référencée. Il est possible de contourner ce problème au prix d'une programmation plus complexe qui n'est pas l'objet de ce document.

Création de la table principale

Nous allons créer une table avec les champs suivants :

- Accounted : Booléen qui indique si la dépense est comptabilisée ou non
- DateAchat : Date qui indique le jour de la dépense
- SommeDépensée : Décimal qui indique le montant de la dépense
- TypeDépense : Texte qui indique la catégorie de la dépense
- ModePaiement : Texte qui indique le moyen de paiement utilisé
- Notes : Texte qui sert de commentaire sur la dépense

Pour cela, nous utilisons l'assistant dans la liste des Taches correspondant aux Tables avec l'exemple de table Dépenses de la catégorie Professionnel. On ajoutera le champ RéfDépenses à la place du champs Accounted qui n'existe pas (on le renommera après).

Attention

L'ordre des champs n'est pas modifiable une fois la table créée (ou du moins, facilement), faites bien attention à fixer l'ordre qui vous convient. L'ajout d'un champs, une fois la table créée, ne se fera qu'en fin de table. Créez une table en mode Ébauche ou utilisez des champs provisoires pour utiliser des champs personnalisés à des emplacements spécifiques.

Faites bien attention de cliquer sur « Suivant » dans les assistants plutôt que sur « Créer » sinon vous manquerez des paramètres.

Choisissez « Boolean » pour le type du champs « RéfDépenses »

Choisissez « Oui » pour le paramètre « Saisie requise » pour tous les champs sauf « Notes ».

Lorsqu'il vous est proposé de définir une clé primaire, cochez la case « Valeur Automatique » pour ne pas vous embêtez avec cette clé. A noter que BASE a besoin d'une clé primaire dans toutes les tables qu'il manipule. Il est donc indispensable de toujours créer une clé primaire pour toutes vos tables !

Créer la table Dépenses en mode modification de l'Ébauche. De là, vous pouvez renommer le champs RéfDépense en Accounted.

Attention

Planifiez les champs et les tables dont vous aurez besoin dans votre base de données sur papier avant de vous lancer à l'aveugle dans la création de tables dans BASE. En effet, BASE est très peu tolérant aux modifications multiples sur des tables/champs existants surtout si vous avez commencé à saisir des données. Je vous conseille de sauvegarder fréquemment à mesure que vous saisissez votre modèle de données.

Ouvrer la table Dépenses par un double clic et saisissez la ligne suivante de donnée :

Case décochée – 01/01/97 – 135,00 – Alimentation – CB – Essai

A noter qu'en mode Ebauche, le champ DateAchat propose dans les Propriétés du Champ de fixer une valeur par Défaut. On serait tenter de vouloir mettre la date du jour, ce qui dans le langage de BASE est CURRENT_DATE. Hélas, ce n'est pas possible, ce champs n'accepte qu'une date fixe.

Le seul moyen d'avoir la date du jour à la saisie d'un nouvel enregistrement est de passer par une macro.

Par contre, on peut fixer les propriétés d'affichage, aussi, fixons les paramétrages suivants :

Champs DateAchat, de la forme JJ/MM/AAAA

Champs SommeDépensée de la forme Monétaire # ##0,00 [€-40C];[RED]-# ##0,00 [€-40C]

A noter qu'en plus de la sauvegarde au niveau de la table, il sera de bon ton d'avoir le réflexe systématique de sauvegarder au niveau de la base globale en elle-même après chaque modification pour éviter des comportements hasardeux ou des pertes de données en cas de plantage de l'application.

Attention

Le format JJ/MM/AAAA est le format d'affichage, à ne pas confondre avec le format de traitement dans les macro BASE qui sera en anglais de la forme DD/MM/YYYY.

Double-cliquez sur la table Dépenses pour l'ouvrir en mode édition et avoir accès par un clic droit sur une colonne au choix « Formatage de colonne... » qui est l'équivalent des propriétés du Champ en mode Ebauche.



A cette table principale, on va ajouter deux tables annexes permettant de valider la saisie des champs TypeDépense et ModePaiement.

Créons une table Catégorie et une table Paiement en mode Ebauche. Chaque table aura un seul champ texte défini en clé primaire.

| | Nom de champ | Type de champ |
|--|--------------------|-------------------|
| | Catégorie | Texte [VARCHAR] |
| | Couper | |
| | Copier | |
| | Supprimer | |
| | Insérer des lignes | |
| | Clé primaire | |

Pour définir la clé primaire, faites un clic droit sur l'entête de la ligne du champ.

Dans la table Catégorie, on saisira pour l'exemple « Alimentation » et « Loisirs »

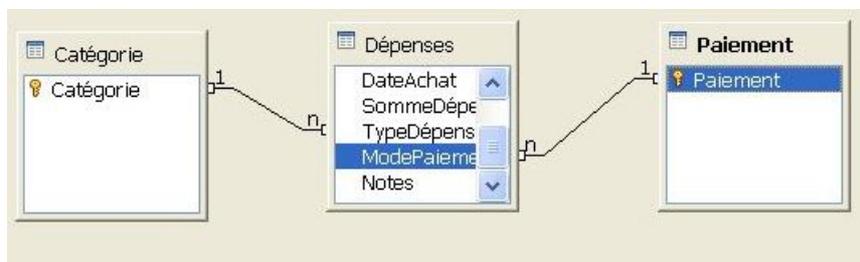
Dans la table Paiement, on saisira pour l'exemple « CB » et « Chèque ».

Afin de contrôler la saisie dans les champs de la table Dépenses, on va créer un lien entre les tables Catégorie et Paiement et la table Dépenses.

Dans le menu Outils → Relations, ajoutez les 3 tables.

Puis cliquer sur le champs Catégorie que vous faites glisser sur le champs TypeDépense de la table Dépenses afin de créer un lien 1:n (le champs Alimentation est utilisé 1 fois dans la table Catégorie pour être utilisé n fois dans la table Dépenses).

Faites de même pour le champs ModePaiement.



Une relation simple mais qui pourrait être beaucoup plus évoluée avec des données complexes et croisées

Ouvrir la table Dépenses et saisissez un enregistrement en vérifiant que vous ne pouvez pas saisir un mode de paiement ou un type de dépense autres que ceux présents dans les tables liés.



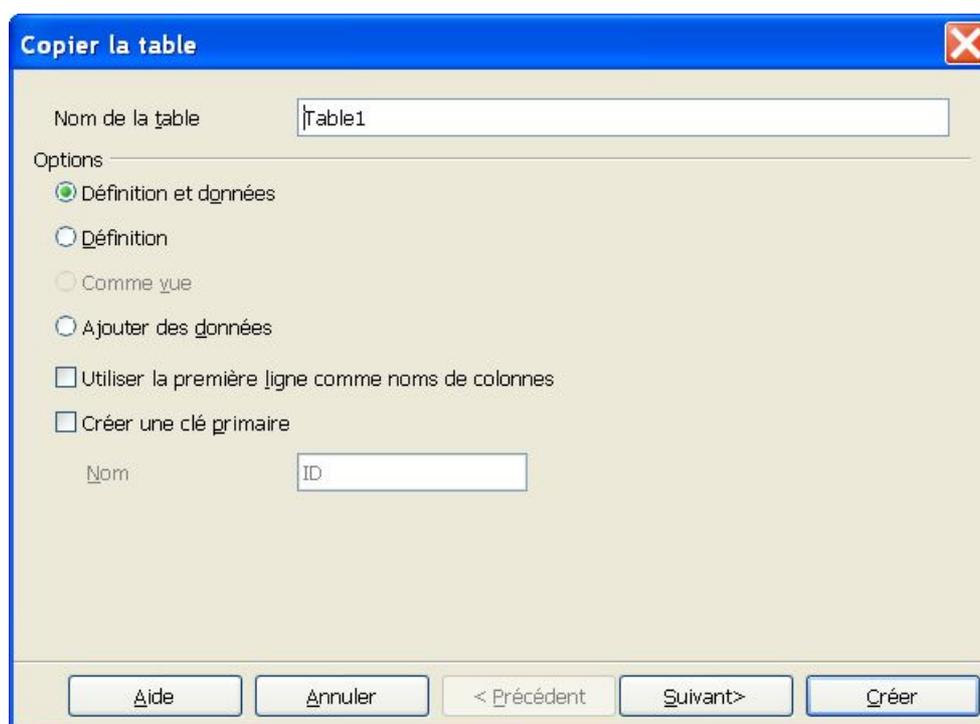
Un message pas très explicite sauf pour les initiés

Maintenant que l'on a défini la structure principale, on va éventuellement importer nos données externes, construire un formulaire pour la saisie de données et définir des requêtes pour automatiser des traitements.

Import / Export

Comme vous pouvez le constater, il n'y a pas de menu Import/Export dans l'application. Cependant, cela est tout à fait possible en passant par l'intermédiaire de Calc.

Pour importer des données dans une table, il suffit de copier les données d'une table Calc puis de coller (clic droit souris) dans l'interface Table de BASE. Cela déclenche un assistant pour importer les données (voir illustration).



On notera le choix peu visible « Ajouter des données » qui permet d'ajouter les données importées à une table existante dans BASE pour peu que l'on prenne soin de spécifier dans le champ « Nom de la table » le même nom que la table cible dans BASE.

Pour exporter des données de BASE, il suffit de glisser une table ou une vue depuis l'interface de BASE sur une feuille Calc à la position A1.

Votre base de données source permet généralement d'exporter dans un format lisible par Calc. De même, Calc permet d'enregistrer dans la plupart des formats supportés pour l'import par les bases de données cibles.

Astuce

Il est déconseillé d'importer vos données à ce stade de la conception. En effet, les temps de traitements pouvant être importants sur des milliers d'enregistrement, je vous recommande de développer vos requêtes et rapports sur un jeu de test d'une dizaine d'enregistrement puis, à la fin, importer vos données.

Formulaire de saisie

On va maintenant créer une interface de saisie de nos données. Cela nous permettra d'avoir une interface (boutons, menus...) et des contrôles sur les champs saisis (listes déroulantes, macros...).

Ce sera beaucoup plus convivial que de passer par l'édition directe de la table.

On lance l'assistant de création de formulaire.

Tout comme pour la création de table, n'hésitez pas à essayer les différents mode de création, à éditer le résultat obtenu, bref à vous accaparer l'interface de BASE.

On sélectionne tous les champs de la table Dépenses sauf la clé primaire ID.

A ce stade, on n'ajoute pas de sous-formulaire qui permet d'afficher les données d'une table liée à un champs (par exemple les détails d'un client sur une facture).

On définit la saisie comme feuille de données.

Les options par défaut conviennent et l'on fait Créer.

Vous noterez que le format d'affichage des champs du formulaire est indépendant du format d'affichage des champs défini au niveau de la table.

Fermez le formulaire et éditez-le (clic droit et choisissez Modifier sur le formulaire Dépenses).

Comme précédemment, faites un clic droit sur les champs pour choisir «Colonne».

Cette fois-ci, vous aurez beaucoup plus de possibilités.

Pour résoudre le problème de la saisie de la date, on va définir à Oui la rubrique Déroulante dans l'onglet Général du paramètre Colonne du champs DateAchat. Pour le format de la date, vous pouvez fixer JJ/MM/AAAA.

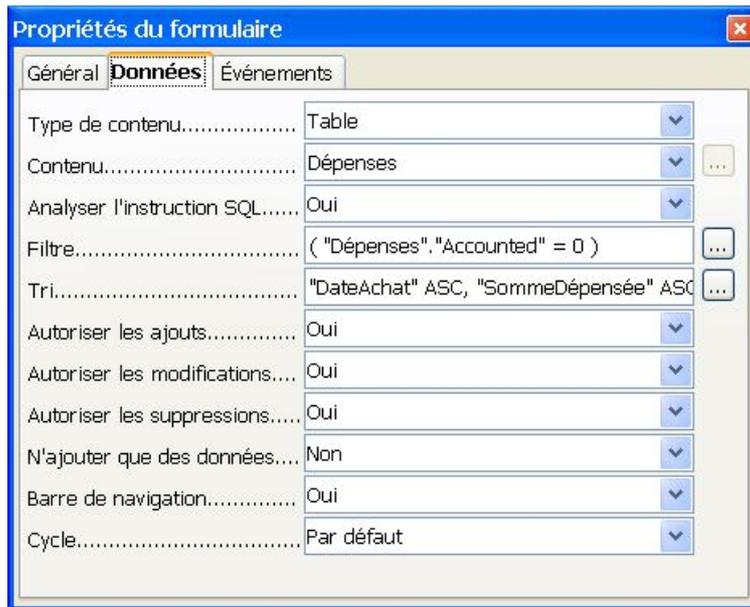


Vous avez maintenant un menu pour la date, ce qui est une alternative simple à la date du jour par défaut.

Choisissez le formatage monétaire pour le champs SommeDépensée.

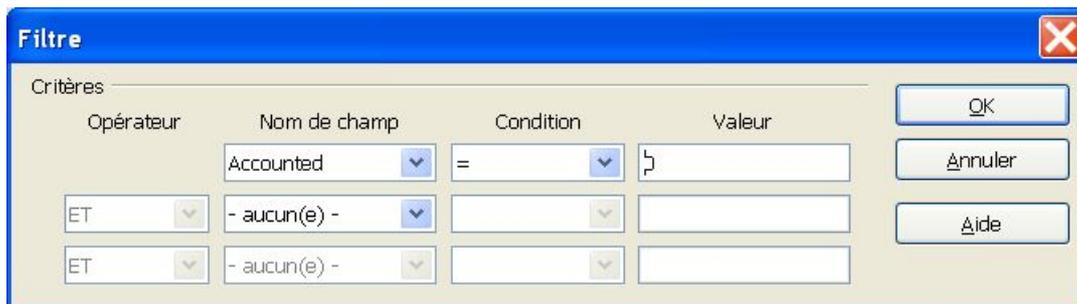
Astuce

En mode édition, faites un clic droit sur la table de donnée et choisissez « Formulaires... », vous aurez accès à l'onglet « Données » qui vous permettra de définir les données sources du formulaire (Table, Requête, SQL) mais aussi d'appliquer en mode graphique des critères de filtrage et de tri.



La source de données de votre table et les critères associés

En cliquant sur les « ... », on accède au filtrage que je vous conseille d'effectuer sur le champs Accounted à la valeur 0 (non comptabilisée) pour la suite de ce pas à pas.



On filtre sur le paramètre Accounted



De même pour le tri

Attention

Mes nombreux tests ont montré une instabilité et des comportements aléatoires si l'on change à répétition la source de données de table à requête puis à SQL, notamment dans la section Rapports. Je vous conseille dans la mesure du possible d'éviter de changer la source de données.

Liste déroulante

Maintenant, il serait utile pour éviter toute frustration de pouvoir saisir le TypeDépense et le ModePaiement à l'aide d'une liste déroulante. En fait, il faut définir que le champ utilisé par le formulaire est une Liste. Par défaut, il est du type du Champ, à savoir un champ texte.

En mode édition, faites un clic droit sur l'entête de colonne TypeDépense du formulaire et choisissez Remplacer par → zone de liste.

Puis faites un clic droit sur l'entête de colonne pour choisir «Colonne » et constater que les choix possibles ont changé.

Allez dans l'onglet Données et choisissez « SQL » (je vous avoue que je n'ai pas compris le fonctionnement de Champs de table qui ne permet pas d'accéder aux champs de la table) dans le type de contenu de liste. Puis saisissez la requête SQL permettant d'obtenir la liste des catégories `SELECT "Catégorie" FROM "Catégorie"`.



Notez que l'on pourrait créer une liste fixe par une saisie à la main des valeurs !

Faites de même pour le champs ModePaiement avec la table Paiement.

Astuce

Dans les paramètres, onglet Général, dans le champs Nombre de lignes, n'hésitez pas à mettre une valeur élevée pour éviter que la liste de choix soit tronquée. De même, n'hésitez pas à changer le nom du champs dans la rubrique Etiquette afin d'avoir un affichage adapté.

Et si on faisait des boutons ?

Pour améliorer les possibilités de notre application, nous allons ajouter des boutons à notre formulaire. Dans la zone d'outils à gauche, sélectionner un bouton et dessinez-le sur le formulaire dans la zone en dessous de la table.

Faites un clic droit sur le bouton et choisissez Contrôle...

De là, explorez les possibilités offertes puis choisissez « NEW » comme étiquette et dans le champs Action, choisissez « Nouvel Enregistrement » parmi la liste des actions possibles.

Vous venez de créer un bouton qui va créer une nouvelle ligne à saisir.
Testez votre formulaire.

Astuce

En faisant un clic droit sur un élément d'interface comme un bouton, je vous conseille de sélectionner « Position et Taille » puis de définir comme paramètres « Ancrer à la page » et de cocher la case « Position » dans la section Protéger afin d'éviter des déplacements hasardeux des éléments d'interface.

Ce serait bien entendu bien mieux si l'on pouvait faire un bouton qui lance une action qui modifie notre table Dépenses. Par exemple, marquer le Accounted à vrai pour tous les enregistrements dont la DateAchat est antérieur au mois en cours.

Que ce soit une requête pour obtenir des données, l'ouverture d'un autre formulaire ou des requêtes qui modifient la table (UPDATE, INSERT), cela devra être lancé par l'appel à une Macro.

Les étapes sont les suivantes :

- Créer une Macro
- Affecter la Macro à un événement

Macro : le B.A BA

Pour des besoins évolués, il faudra vous plonger dans la documentation de BASE, lire les forums et les nombreuses aides. L'idée, ici, est de vous apprendre le minimum requis pour faire face aux besoins de base.

Un très bon document de référence pour les Macros est « Getting started with Macros »⁷

Aller dans Outils → Macros → Gérer les Macros → LibreOffice Basic...

Puis sélectionner votre base de données, choisissez Nouveau et valider pour commencer à créer des macros dans un module de votre base de données !

Ouvrir un rapport / Fermer un rapport

```
Sub Ouvrir_Rapport
```

```
  ThisDatabaseDocument.ReportDocuments.getByName ("Nom_du_Rapport").open
```

```
  ' Pour fermer un rapport, on utilisera le terme close au lieu de open
```

```
End Sub
```

Astuce

N'oubliez pas qu'un rapport peut être basé sur une source de données qui est une vue, une requête ou du code SQL. Le lancement d'un rapport permet donc d'avoir un affichage riche en terme de possibilités sans avoir besoin de traiter un appel SQL au sein de la macro.

Ouvrir un formulaire / Fermer un formulaire

```
Sub Ouvrir_Formulaires_Depenses
```

```
  ThisDatabaseDocument.FormDocuments.getByName ("Nom_du_formulaire").open
```

```
  'Pour fermer un formulaire, on utilisera le terme close au lieu de open
```

```
End Sub
```

Astuce

N'oubliez pas qu'un formulaire peut être basé sur une source de données qui est une vue, une requête ou du code SQL. Le lancement d'un formulaire permet donc d'avoir un affichage riche en terme de possibilités sans avoir besoin de traiter un appel SQL au sein de la macro.

⁷ http://wiki.documentfoundation.org/cgi_img_auth.php/0/07/0113GS3-GettingStartedWithMacros.pdf

Exécuter une requête SQL (Select, Update ou Insert)

```
Sub Executer_SQL
```

```
Dim oDBContext As Object , oDB As Object , oBase As Object
```

```
Dim oStatement As Object , oRequete As Object
```

```
Dim strSQL As String
```

```
'On se connecte à la base de données référencées
```

```
oDBContext = CreateUnoService("com.sun.star.sdb.DatabaseContext")
```

```
oDB = oDBContext.getByName("NOM_DE_VOTRE_BASE_DE_DONNEES")
```

```
'La base est sans login/password
```

```
oBase = oDB.getConnection("", "")
```

```
'On prépare la requête
```

```
oStatement = oBase.createStatement()
```

```
'On saisit la requête que l'on stocke dans la chaine strSQL
```

```
'A noter, chaque " doit être doublé et la requête SQL doit être entre ".
```

```
strSQL = "UPDATE ""Dépenses"" SET ""Accounted""= 1 WHERE  
(YEAR("""DateAchat"")) < YEAR(CURRENT_DATE)) OR ((MONTH("""DateAchat"")) <  
MONTH(CURRENT_DATE)) AND (YEAR("""DateAchat"")) = YEAR(CURRENT_DATE))"
```

Astuce

Pour partir d'une base de langage SQL, vous pouvez utiliser le mode « Query Design » en sélectionnant dans le mode « Requetes » la tâche « Créer une requête en mode Ebauche... ». Une fois la requête définies en mode graphique assisté, vous la sauvegardez puis par un clic droit sur la requête vous choisissez « Editez en mode SQL ».

```
'On exécute la requête SQL
```

```
oRequete = oStatement.executeQuery( strSQL )
```

```
'On ferme proprement les appels
```

```
oRequete.Close
```

```
oStatement.Close
```

```
oBase.Close
```

```
oBase.Dispose
```

```
End Sub
```

Attention

Si on lance une requête qui modifie une table affichée dans un formulaire, l'affichage ne sera mis à jour qu'après un rafraîchissement de l'affichage du formulaire. Il convient donc de faire une Macro qui gère ce rafraîchissement si cela est pertinent.

Astuce

Pour construire et tester votre requête SQL avant d'écrire votre macro, vous pouvez utiliser le menu Outils → SQL... qui accepte toutes les commandes SQL.

Comme vous l'avez remarqué, si vous avez testé le mode Requêtes de BASE, il ne permet pas de faire autre chose que l'équivalent d'un SELECT en SQL.

Et oui, il n'est pas possible de mettre à jour une table ou d'insérer des enregistrements dans une table en utilisant un assistant graphique, il faudra obligatoirement passer par la saisie d'une requête SQL et l'exécution d'une Macro !

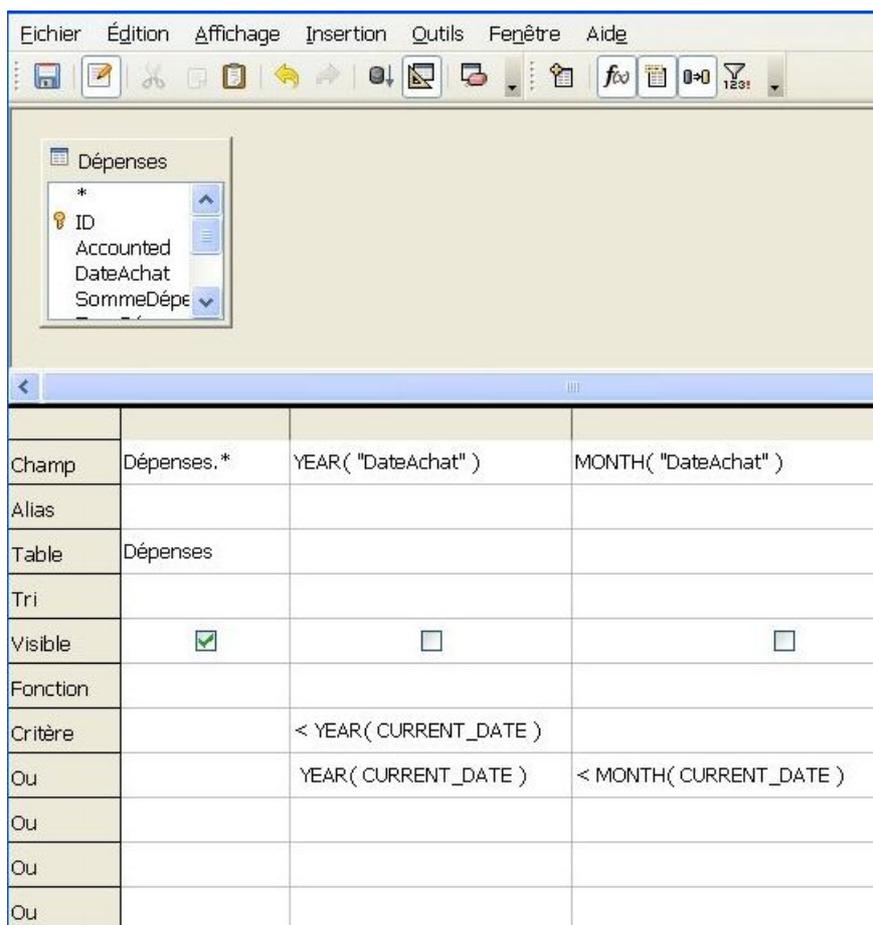
Update et Insert en mode graphique

Même si BASE est limité aux requêtes SELECT en mode graphique, cela permet cependant de faciliter le travail en vue de construire sa requête UPDATE ou INSERT.

Prenons l'exemple de la mise à jour dans notre table Dépenses du champ Accounted pour les enregistrements dont la DateAchat est inférieur au mois en cours.

Pour construire cette requête complexe, on peut passer par les étapes suivantes :

1- Création d'une requête en mode ébauche où l'on va sélectionner les enregistrements dont la DateAchat est inférieur au mois en cours.



| Champ | Dépenses.* | YEAR("DateAchat") | MONTH("DateAchat") |
|----------|-------------------------------------|--------------------------|--------------------------|
| Alias | | | |
| Table | Dépenses | | |
| Tri | | | |
| Visible | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Fonction | | | |
| Critère | | < YEAR(CURRENT_DATE) | |
| Ou | | YEAR(CURRENT_DATE) | < MONTH(CURRENT_DATE) |
| Ou | | | |
| Ou | | | |
| Ou | | | |

On n'est pas obligé de faire la requête complète si elle est complexe, l'idée est d'avoir un support de code SQL

2- Sur notre base de test, on s'assure d'avoir des dépenses avec DateAchat sur le mois en cours et des enregistrements avec une date d'achat antérieure et postérieure au mois en cours. Puis on lance la requête.

| ID | Accounted | DateAchat | SommeDépensée | TypeDépense | ModePaiement | Notes |
|-------|--------------------------|------------|---------------|--------------|--------------|-------|
| 0 | <input type="checkbox"/> | 01/01/1997 | 135,00 € | Alimentation | CB | Essai |
| 7 | <input type="checkbox"/> | 01/01/1998 | 245,00 € | Alimentation | CB | eee |
| 16 | <input type="checkbox"/> | 01/01/1998 | 255,00 € | Loisirs | CB | dd |
| 19 | <input type="checkbox"/> | 15/03/2011 | 45,00 € | Alimentation | CB | ddd |
| 21 | <input type="checkbox"/> | 28/05/2011 | 20,00 € | Alimentation | CB | eee |
| 22 | <input type="checkbox"/> | 01/02/2011 | 12,00 € | Alimentation | CB | fff |
| <Auto | <input type="checkbox"/> | | | | | |

La base de test

| ID | Accounted | DateAchat | SommeDépensée | TypeDépense | ModePaiement | Notes |
|-------|--------------------------|------------|---------------|--------------|--------------|-------|
| 0 | <input type="checkbox"/> | 01/01/1997 | 135,00 € | Alimentation | CB | Essai |
| 7 | <input type="checkbox"/> | 01/01/1998 | 245,00 € | Alimentation | CB | eee |
| 16 | <input type="checkbox"/> | 01/01/1998 | 255,00 € | Loisirs | CB | dd |
| 22 | <input type="checkbox"/> | 01/02/2011 | 12,00 € | Alimentation | CB | fff |
| <Auto | <input type="checkbox"/> | | | | | |

Le résultat de notre requête validée

2- En éditant en mode SQL la requête créée graphiquement, on obtient :

```
SELECT * FROM "Dépenses" WHERE ( YEAR( "DateAchat" ) <
YEAR( CURRENT_DATE ) OR YEAR( "DateAchat" ) = YEAR( CURRENT_DATE ) AND
MONTH( "DateAchat" ) < MONTH( CURRENT_DATE ) )
```

3- Sur la base de ce code SQL, il ne reste plus qu'à remplacer le SELECT par un INSERT, de doubler les " et éventuelle ajouter des parenthèses si on préfère séparer les clauses :

```
SELECT * FROM "Dépenses" WHERE ( YEAR( "DateAchat" ) <
YEAR( CURRENT_DATE ) OR YEAR( "DateAchat" ) = YEAR( CURRENT_DATE ) AND
MONTH( "DateAchat" ) < MONTH( CURRENT_DATE ) )
```

devient donc

```
UPDATE ""Dépenses"" SET ""Accounted""= 1 WHERE (YEAR(""DateAchat"") <
YEAR(CURRENT_DATE)) OR ((MONTH(""DateAchat"") < MONTH(CURRENT_DATE)) AND
(YEAR(""DateAchat"") = YEAR(CURRENT_DATE)))
```

4- On peut insérer cette requête dans notre macro !

Deux macros spécifiques à notre projet

Tant qu'à faire, écrivons la macro qui va mettre à jour le champs Accounted et qui va rafraîchir le formulaire qui affiche les champs de la table Dépenses. On suppose que vous avez enregistré votre base avec le nom « comptes » :

```
Sub MarquerAccountedAll (oEvent as object)
Dim oForm As Object
Dim oDBContext As Object , oDB As Object , oBase As Object
Dim oStatement As Object , oRequete As Object
Dim strSQL As String

'On récupère l'objet Formulaire
oForm=oEvent.source.model.parent

oDBContext = CreateUnoService("com.sun.star.sdb.DatabaseContext")
oDB = oDBContext.getByName("comptes")

oBase = oDB.getConnection("", "")
oStatement = oBase.createStatement()

'Mise à jour d'enregistrements:
strSQL = "UPDATE ""Dépenses"" SET ""Accounted""= 1 WHERE
(YEAR("""DateAchat"")) < YEAR(CURRENT_DATE)) OR ((MONTH("""DateAchat"")) <
MONTH(CURRENT_DATE)) AND (YEAR("""DateAchat"")) = YEAR(CURRENT_DATE))"
oRequete = oStatement.executeQuery( strSQL )
'-----

'On rafraîchit l'affichage
oForm.Reload

oRequete.Close
oStatement.Close
oBase.Close
oBase.Dispose
End Sub
```

Tant qu'à faire, on va en profiter pour faire une macro qui écrit une Dépense récurrente de façon automatique :

```
Sub Ecritures_mensuelles (oEvent as object)
Dim oForm As Object
Dim oDBContext As Object , oDB As Object , oBase As Object
Dim oStatement As Object , oRequete As Object
Dim strSQL As String

'On récupère l'objet Formulaire
oForm=oEvent.source.model.parent

oDBContext = CreateUnoService("com.sun.star.sdb.DatabaseContext")
oDB = oDBContext.getByName("comptes")

oBase = oDB.getConnection("", "")
oStatement = oBase.createStatement()

'Ajout d'écritures dans la tables Dépenses

'On ajoute un prélèvement automatique à la date du jour
' On prendra soin d'ajouter la catégorie Charges Mensuelles Fixes dans
la table Catégorie
' On prendra soin d'ajouter le paiement Prélèvement Automatique dans la
table Paiement
strSQL = "INSERT INTO ""Dépenses"" (""Accounted"", ""DateAchat"",
""SommeDépensée"", ""TypeDépense"", ""ModePaiement"", ""Notes"") VALUES
(0, CURRENT_DATE, 10, 'Charges Mensuelles Fixes', 'Prélèvement
Automatique', 'Entrée automatique : Donation à la Document Foundation')"
oRequete = oStatement.executeQuery( strSQL )

strSQL = "INSERT INTO ""Dépenses"" (""Accounted"", ""DateAchat"",
""SommeDépensée"", ""TypeDépense"", ""ModePaiement"", ""Notes"") VALUES
(0, CURRENT_DATE, 400, 'Charges Mensuelles Fixes', 'Prélèvement
Automatique', 'Entrée automatique : Mon loyer')"
oRequete = oStatement.executeQuery( strSQL )

'On rafraîchit l'affichage
oForm.Reload
```

```
oRequete.Close
oStatement.Close
oBase.Close
oBase.Dispose
End Sub
```

Attention

Pour des raisons propre à BASE, votre fichier de base de données peut grossir au fil du temps (un peu comme un fichier Word). Il existe une requête SQL à lancer par le menu ou via une macro qui est "CHECKPOINT DEFRAG". Cela permet à la base de subir une cure de dégrossissement après de nombreuses manipulations. Pensez-y !

Assigner une Macro

Maintenant que nous avons écrit des Macros, nous allons les utiliser. Retournons dans notre formulaire Dépenses en mode édition et ajoutons deux boutons, l'un que l'on nommera « Marquer comptabilisé » et l'autre « Initialisation Dépenses du Mois ».

Puis faites un clic droit sur le bouton, choisissez « Controles... » puis l'onglet Evénements, et en face du champs « Exécuter l'action » cliquer sur les « ... » pour affecter une Macro, en l'occurrence la Macro « Ecritures_mensuelles » sur le bouton d'écriture mensuelles et de même pour le bouton « Initialisation Dépenses du Mois » avec la macro « MarquerAccountedAll ».

Testez le fonctionnement des boutons

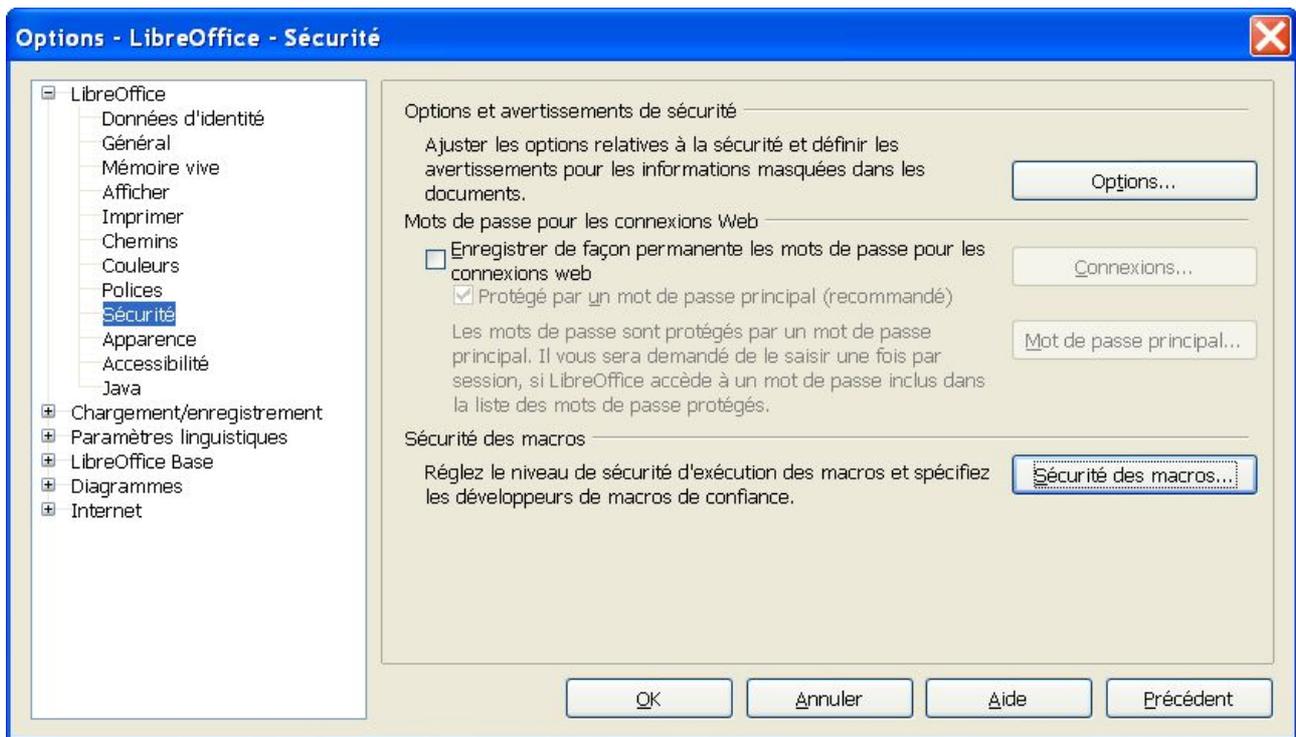
Vous pouvez aussi créer un bouton « Quit » qui lancera une macro qui fermera le formulaire « Dépenses » par exemple.

Ou encore faire un bouton qui va lancer un rapport de toutes les dépenses d'un mois donné...

La seule limite est votre imagination.

Attention

Par défaut, BASE est configuré pour interdire le chargement des Macros au lancement d'un fichier de base de données. Pour éviter tout problème, modifier les paramètres de sécurité ou attribuer une zone de sécurité pour le lancement de votre base de données.



Dans Outils → Options sur PC ou Préférences sur Mac, on réglera la sécurité à moyen pour les macros ou on définira une zone de confiance

Lancement d'une macro à l'ouverture d'un document BASE

Un cas particuliers est le lancement de Macro à l'ouverture de votre fichier de bases de données, ce qui est très pratique pour ouvrir par défaut une interface utilisateur...

La macro pour ouvrir un formulaire au lancement est la suivante :

```
Sub Menu_Principal
```

```
    'Il faut lancer la connexion à la base de données
```

```
    ThisDatabaseDocument.CurrentController.connect("", "")
```

```
    'Enfin l'appel standard peut avoir lieu
```

```
    ThisDatabaseDocument.FormDocuments.getByName("Dépenses").open
```

```
End Sub
```

Les Rapports

Enfin, notre exemple ne serait rien sans rapport pour avoir un état synthétique de ses données.

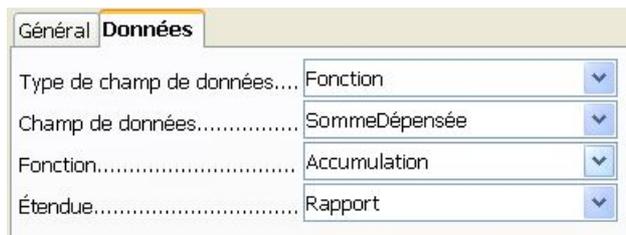
En utilisant l'assistant pour créer un rapport, ce qui est bien pratique, on sera souvent insatisfait du résultat et l'on souhaitera ajouter des groupements, modifier le filtrage ou la source de données ultérieurement.

Certaines actions sont peu intuitives. Par défaut, le groupement est défini avec un en-tête mais pas de pied de section. Aussi, si vous voulez ajouter une somme cumulée pour un regroupement, il vous faudra ajouter le pied de section en passant par le menu : Affichage -> Tri et groupement



Ici, vous pouvez paramétrer le groupement et notamment le pied de page

A partir de là, vous pouvez ajouter un champs dans le pied de section (icone de la barre d'outils ou raccourci Alt-F8, bizarrement, le menu insertion ne permet pas l'ajout d'un champs) et en faisant un clic droit dessus puis Propriétés, accéder aux Données du champ.



Ici, on a défini le champ comme une fonction accumulation sur tout le rapport pour afficher le montant total des sommes dépensées affichées par le rapport.

Attention

Par défaut, BASE est très instable sur l'ajout d'un champ faisant une accumulation sur un ensemble de données groupées, surtout si vous avez plusieurs niveaux d'accumulation et de groupements! Je vous recommande de sauvegarder fréquemment (le rapport et la base) avant tout manipulation.

Paradoxalement, si vous voulez accéder aux propriétés globales du rapport, il vous faudra transiter par des propriétés locales. Cela a lieu en 2 étapes :

- 1- Faites un Clic droit sur une zone du rapport pour accéder aux Propriétés de la zone
- 2- Cliquer sur la zone grise en dehors du rapport
- 3- Vous avez ainsi accès aux propriétés globales du rapport

3 Le mode propriétés du rapport s'affiche alors!

Outre la source de données, vous aurez accès aux filtres et au format de sortie du rapport

Attention

Étonnamment, la source de données peut différer de celle donnée à l'assistant (ex : la source d'une table Requête va être remplacé par une source SQL équivalent à la requête). Vérifier bien cela dans votre rapport car sinon la modification de la source de données (ex : une table Requête) n'entraînera pas de modification des données affichées dans le rapport qui se base sur une requête SQL autonome par exemple.

La zone filtre est très pratique, car elle va vous permettre de générer un rapport avec un paramètre inconnu à saisir au moment de la génération du rapport.

Pour cela, au lieu d'écrire une valeur comme paramètre, il suffit de respecter la syntaxe « : » suivi du nom de la variable qui fera l'objet d'une interrogation.

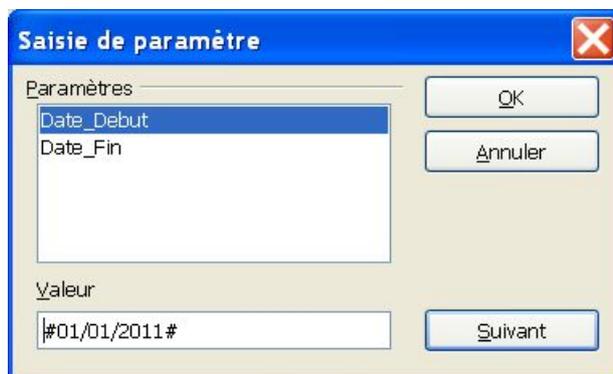
Attention

Le nom de la variable ne doit pas comporter d'accents, d'espace ou de caractères spéciaux. Ainsi **:DateAnnee** est valide alors que **:Date Année** est invalide. Il est possible aussi d'utiliser par défaut le caractère ? qui se contentera d'afficher le nom réel du champ objet du filtrage.

| | | | | | | | | |
|----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-----------|
| Champ | YEAR([DateA | TypeDépense | SommeDépensée | Somme("Somme | YEAR("DateAchat | YEAR("DateAchat | | |
| Alias | Année | TypeDépense | Total Dépenses | Moyenne Mensuel | | | | |
| Table | | Dépenses | Dépenses | | | Dépenses | | |
| Tri | | croissant | | | | | | |
| Visible | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| Fonction | | Groupe | Somme | | | Groupe | | |
| Critère | | | | | | | | >= :Annee |
| Ou | | | | | | | | |
| Ou | | | | | | | | |

Par exemple, on peut filtrer sur une plage de date à saisir par l'utilisateur via les propriétés Filtre du rapport.

Si la source est une requête ou du code SQL, on peut très bien procéder de même en mettant:variable à l'endroit désiré.



Lors de l'exécution, on aura alors une boîte de dialogue qui demandera les paramètres. Pas besoin de s'embêter avec les caractères # qui seront ajoutés automatiquement. Il faut juste veiller à saisir le bon format du paramètre sous peine d'erreur (une date complète et pas un chiffre ou des

lettres si une date est requise). Le mode de requête est assez rustique et sans possibilité de messages personnalisés mais il a l'avantage d'être simple et rapide à mettre en œuvre

Astuce

Lors de l'impression de votre rapport, certains champs, label texte ou autres données peuvent apparaître tronqués. C'est notamment le cas pour les en-têtes et pieds de page qui peuvent se retrouver à cheval sur 2 pages. Je vous conseille dans les propriétés de ces sections de spécifier à « Oui » le paramètre « Conserver ensemble ».

| Général | |
|--|------------------------|
| Nom..... | Grouper le pied de pag |
| Forcer une nouvelle page..... | Aucun(e) ▼ |
| Conserver ensemble..... | Oui ▼ |
| Répéter la section..... | Non ▼ |
| Visible..... | Oui ▼ |
| Hauteur..... | 1,51cm ▲▼ |
| Expression d'impression conditionnelle.... | ▼ ... |
| Transparence d'arrière-plan..... | Non ▼ |
| Couleur d'arrière-plan..... | Jaune pâle ▼ |

On notera que, les en-têtes/pieds de page dans un rapport (que ce soit au niveau d'un groupe ou du rapport global) permettent des traitements sophistiqués avec les expression d'impression conditionnelle.

Conclusion

Voilà, j'espère que ce document vous aura permis de démarrer avec BASE pour vos propres besoins personnels en terme de base de données. N'oubliez pas que les exemples données n'ont aucune prétention de respecter des normes ou de mettre en avant des méthodologies. Il s'agissait simplement d'illustrer le principe de fonctionnement de BASE et de vous donner des clés pour démarrer !