

# **OpenOffice.org et les bases de données.**

*Diffusé par*

*Le Projet Documentation OpenOffice.org*

## Table des Matières

Introduction :.....	3
Présentation de la base :.....	3
Migrer une base de données MS Access vers PostgreSQL.....	4
Utiliser une base de données PostgreSQL avec OpenOffice.org.....	9
Première solution :.....	11
Deuxième solution :.....	11
Migrer une base de données MS Access vers un format DBF.....	22
Utiliser un format Dbase sous OpenOffice.org.....	22
Créer un formulaire et un sous formulaire avec des fichiers DBF.....	23
Fin de notre itinéraire :.....	27
Crédits .....	28
Licence.....	28

## Introduction :

### *Ce que vous trouverez dans ce document :*

Comment migrer une base de données MS Access vers PostgreSQL  
 Comment utiliser une base de données PostgreSQL avec OpenOffice.org (lien ODBC)  
 Comment migrer une base de données MS Access vers un format Dbase (.dbf)  
 Comment créer des formulaires et des sous formulaires avec OpenOffice.org.

### *Ce que vous ne trouverez pas :*

Comment installer, configurer et administrer un serveur de base de données MySQL ou PostgreSQL sous GNU/Linux ou sous Windows.  
 Comment installer et configurer le pilote ODBC sous GNU/Linux.

Ce document n'a pas vocation à être exhaustif, ni réellement pédagogique. Il a été conçu à la demande d'Yves POTIN, afin d'aider les personnes qui passent d'un existant sous Windows vers la Debian Éducation.

Si vous avez d'autres éléments qui peuvent améliorer ou compléter ce document, même au brouillon, merci de nous les faire parvenir.

## Présentation de la base :

Voici le schéma de la base qui sert à illustrer le document :

The screenshot shows the OpenOffice.org interface with a database table and its schema diagram. The table has the following data:

cla_id	cla_nom	ele_id	ele_nom	cla_id1
1	Quatrième 4	1	Marie DATOUY	1
1	Quatrième 4	2	Jean DUGRAND	1
2	Troisième 3	3	Andriati MARCHAVELLO	2

The schema diagram shows two tables: 'classe' and 'eleve'. The 'classe' table has fields 'cla\_id' and 'cla\_nom'. The 'eleve' table has fields 'ele\_id', 'ele\_nom', and 'cla\_id'. A line connects the 'cla\_id' field in the 'eleve' table to the 'classe' table, indicating a foreign key relationship.

Ici le schéma est représenté sous OpenOffice.org, mais c'est la même structure qui a été utilisé sous MS Access et sous PostgreSQL.

Peu importe les types de données que vous prendrez, la seule chose qui est importante est que la rubrique `cla_id` de classe soit de même type que `cla_id` de eleve.

## Migrer une base de données MS Access vers PostgreSQL

### Voici ce que l'on désire réaliser :

- On dispose d'une base de données sous MS Access, et on désire porter tout ça sur une base de données PostgreSQL sous Linux. On utilisera une exemple simple avec 3 tables (une table classe, une table élève qui contient les informations des élèves, une table note).
- On dispose de 2 machines en réseau. La première sous Windows avec Access, la seconde avec la Debian Éducation et PostgreSQL.

Note : Le choix PostgreSQL sous Linux est volontaire, mais réaliser les manipulations vers une base Postgres sous Windows ou vers une base MySQL est tout aussi envisageable. Des notes pour cela seront données à la fin du document.

Vous devez avoir sur votre machine Linux les produits suivants d'installés :

- Le serveur de base de données PostgreSQL
- Le pilote ODBC pour PostgreSQL

Le pilote ODBC servira à l'utilisation des bases de données à partir d'OpenOffice.org.

Vous devez avoir sur votre machine Windows les produits suivants d'installés :

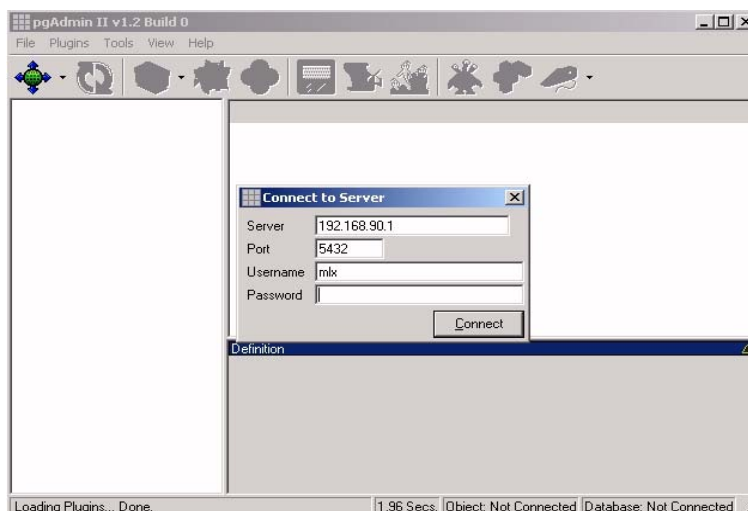
- L'interface d'administration des Bases PostgreSQL pgADMIN
- L'interface de transformation d'une base MS Access vers PostgreSQL
- Le pilote ODBC pour Postgres sous Windows.

Ces produits sont téléchargeables sur : <http://www.pgadmin.org> et sur <http://odbc.postgresql.org>.

Les deux machines doivent communiquer sur le réseau.

### Préparation de la machine windows

- 1 - Téléchargez et installez le pilote ODBC.
- 2 - Téléchargez et installez pgAdmin.
- 3 - On va utiliser pgAdmin pour créer une nouvelle base de données sur la machine Linux. Ouvrez d'abord la connexion :



Vous mettez l'adresse de votre serveur Linux, le port (par défaut 5432), le compte utilisateur. Ce doit être un compte d'administrateur de base de données.

4 - Pour créer une base de données, faites un Click DR droit sur « databases », « create object », « database », mettez par exemple « eleve ».

Si vous rencontrez des difficultés à ce niveau là, il faut vérifier que :

- le compte que vous utilisez est bien un compte d'administrateur de base de données.
- vous avez les permissions d'accéder au serveur de base de données PostgreSQL à partir du client windows. Cela est souvent défini dans le fichier : pg\_hba.conf.

Voici ce que ça donne par exemple sur ma machine :

*Le fichier est : /var/lib/pgsql/data/pg\_hba.conf*

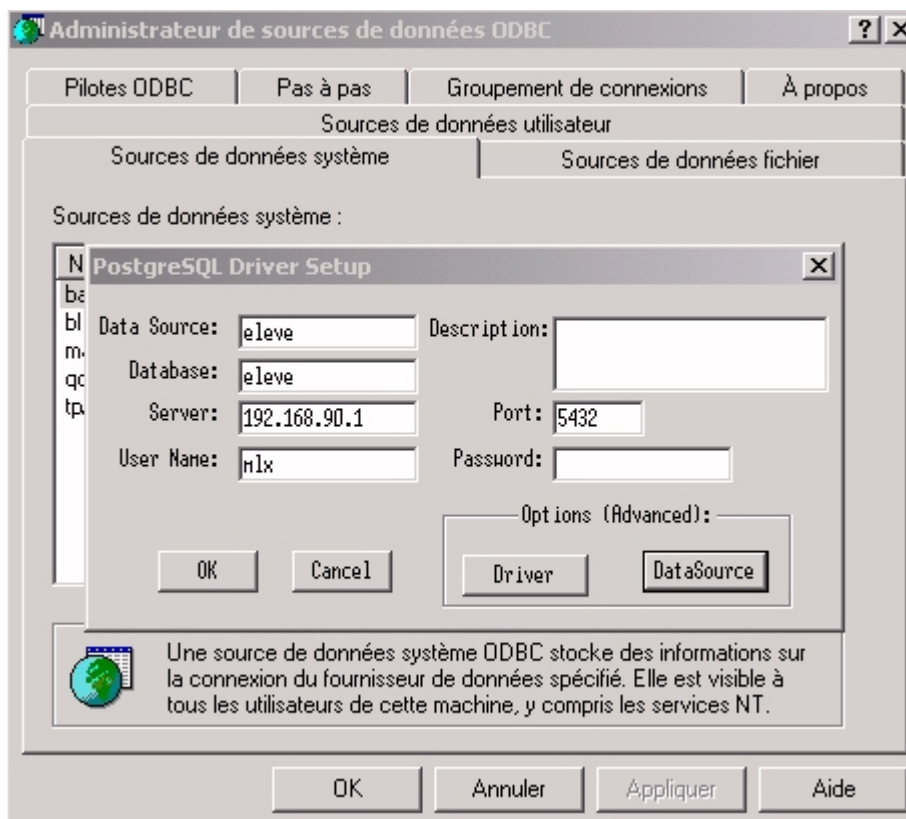
```
host all 192.168.90.0 255.255.255.0 trust
```

pour un accès aux machines de mon réseau (192.168.90.0) sur toutes les bases (all) de mon serveur.

La base de données vierge est créée. Nous allons créer une ressource ODBC pour cette base. En général les outils d'administration ODBC sont accessibles dans le panneau de configuration. Sur la machine dont je dispose (Windows 2000 Server) ils sont dans « panneau de configuration », « outils d'administration ».

5 - Activez « sources de données ODBC »

Prenez, « source de données système » (ou DSN système), « ajouter ». Dans la liste prendre « PostgreSQL » qui doit apparaître, « terminer ».



Data Source : correspond au « nom » que vous donnez à la ressource ODBC que vous déclarez. Vous pouvez l'appeler comme vous voulez.

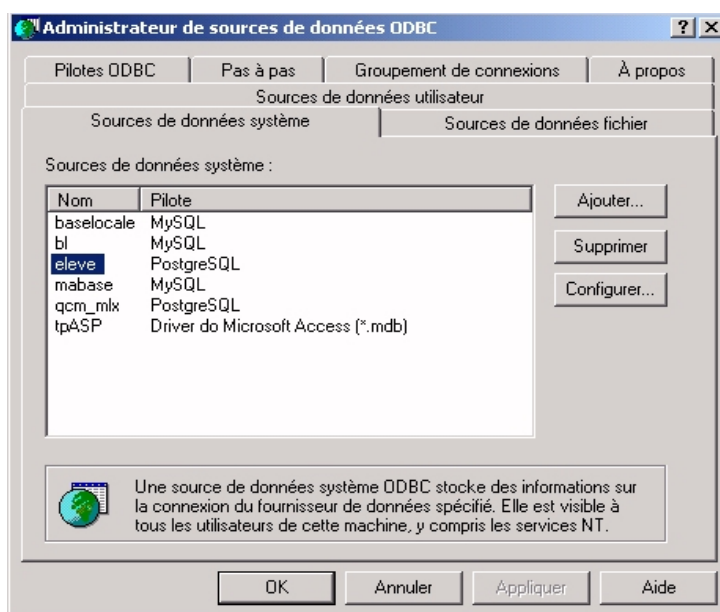
Database : C'est le nom de la base que l'on a créé sur Linux. Là, il faut respecter le nom que vous avez donné.

Server, c'est l'adresse IP ou le nom de votre machine Débian Éducation.

User Name: le compte d'accès à la base de données.

Faites « OK »

La ressource, ici « eleve » doit apparaître dans la liste des sources de données système.



Nous allons tout de suite vérifier que l'accès ODBC est correctement déclaré.

6 - Lancer MS Access, nouvelle base de données.

Allez ensuite dans « fichier », « données externes », « lier des tables »

Prenez « odbc databases » dans « types de fichier »

Prenez « eleve » (votre ressource ODBC) dans « sources de données machines »

Bien sûr vous ne verrez aucune table, puisque votre base distante est vide, mais normalement vous ne devez avoir aucun message d'erreur. Dans le cas contraire, vérifiez que vous ne vous êtes pas trompé dans un paramètre de la ressource ODBC.

### **Migration des tables, première solution :**

Ici on va utiliser la fonction « exporter » de MS Access et une liaison ODBC vers PostgreSQL

- Ouvrez votre base de données élève, MS Access. (attention, ici vous ouvrez eleve.mdb)
- Sur la table « classe », faites un « Click DR », puis exporter
- Dans « type de fichier » prenez « ODBC Databases »
- Entre le nom de la table destination
- Sélectionnez votre source de données ODBC (eleve), validez.

Structures et données, sont transférées sur PostgreSQL.

Si vous avez des messages d'erreur, cela peut provenir :

- des caractères accentués ou des espaces sur les noms de tables ou les noms de champs
- sur certains formats de données date/heures, mal interprétés par PostgreSQL

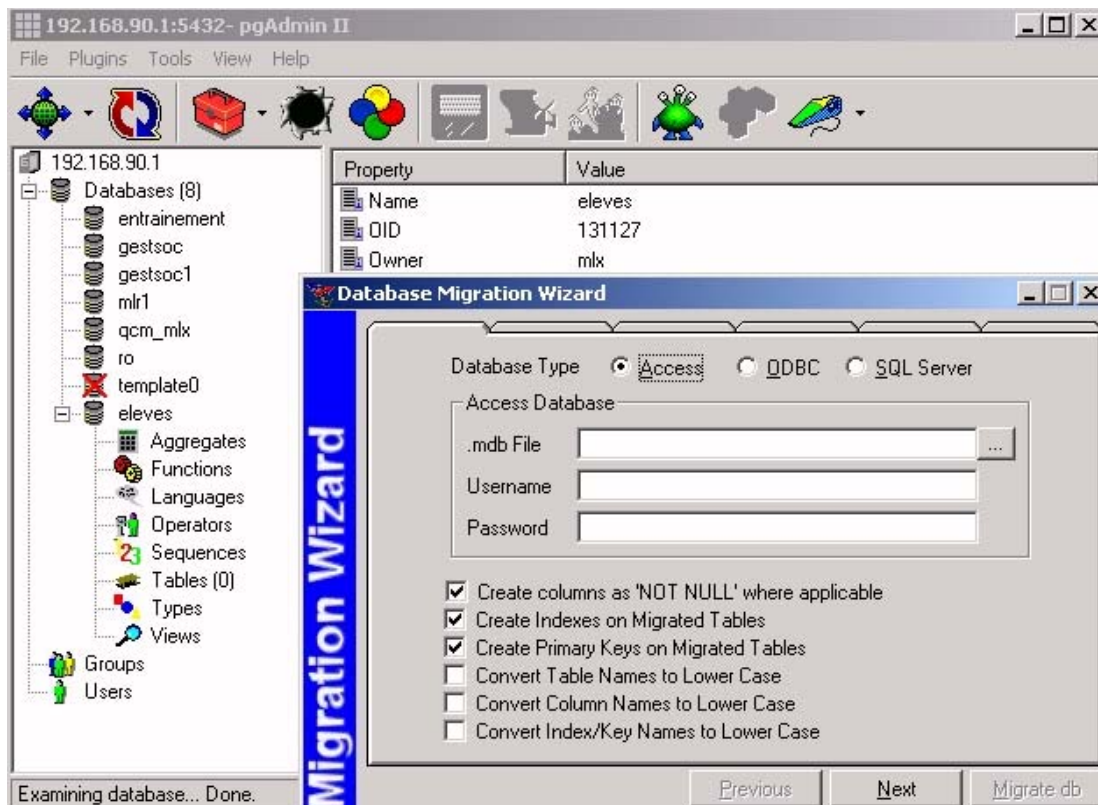
Si c'est votre cas, faites les modifications sur une copie de sauvegarde de votre base de données et réitérez.

### **Migration des tables, deuxième solution :**

Ici on va utiliser un utilitaire intégré à pgAdmin.

Cette option vous permet également d'exporter des informations comme les clés primaires, les clés étrangères de MS Access vers PostgreSQL. Elle permet également le portage de base MS SQL/Server ou MySQL vers PostgreSQL.

- Téléchargez et décompressez pgMigration à partir de <http://www.pgadmin.org> dans le répertoire « plugins » de pgAdmin. Sur ma machine cela donne « C:\Program Files\pgAdmin2\Plugins ». C'est une DLL.
- Lancer pgAdmin et connectez-vous à votre serveur de base de données.
- Par le menu « tools », « options », « plugins », ajoutez « pgMigration.dll », validez.
- Vous avez maintenant dans le menu « plugins », l'option « Data Migration Wizard ». Les onglets seront accessibles au fur et à mesure des étapes de la migration.



- Sélectionnez votre base de données access, next
- Sélectionnez la base PostgreSQL vers laquelle vous désirez effectuer la migration, next
- Sélectionnez les tables à migrer, next
- Sélectionnez le table dont vous voulez exporter également les données, next
- Sélectionnez éventuellement les clefs primaires et étrangères, next
- Validez « migrate db »

Vous pouvez avoir des erreurs liées à la structure des bases access ou des formats de données, mais vous avez une table qui vous permet de « mapper » (établir une table de correspondance) entre des formats de données MS Access et ceux de PostgreSQL.

## Utiliser une base de données PostgreSQL avec OpenOffice.org

Nous allons voir comment utiliser la base de données PostgreSQL « eleve » à partir d'OpenOffice.org. Nous verrons après, comment réaliser un formulaire simple, puis un sous-formulaire. Que vous soyez sous Windows ou sous Linux, vous devez avoir le pilote ODBC d'installé et une ressource ODBC déclarée dans le fichier de configuration.

La déclaration sous Windows a déjà été vue. Voici un exemple sous Linux.

### Déclarer une ressource ODBC sous Linux :

```
#Fichier ~/.odbc.ini
[ODBC Data Sources]
```



```
#Ici vous déclarez vos ressources, vous pouvez mettre n'importe quel nom
qcm = Read/Write Database
eleves = Read/Write Database
```

```
# Ici vous déclarez les propriétés de chaque ressources
[qcm]
ReadOnly = 0
Servername = localhost
Port = 5432
Database = qcm_mlx
Driver = /usr/lib/libodbcpsql.so
Username = mlx
```

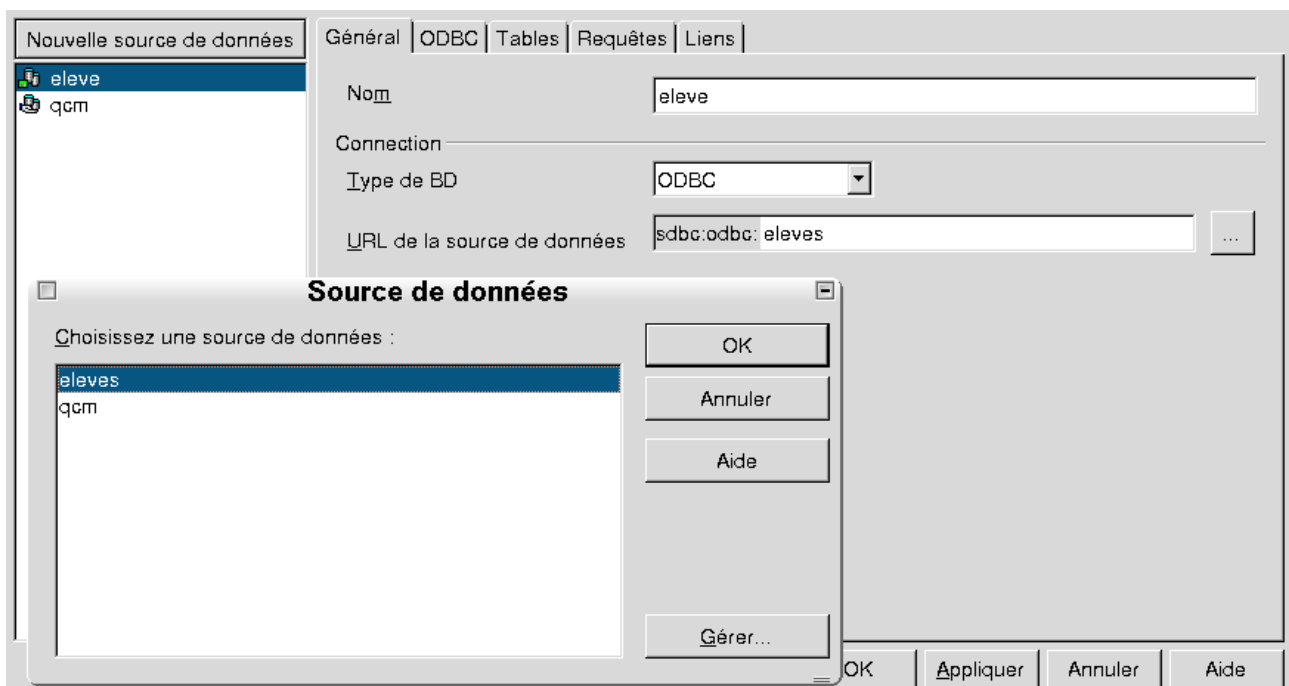
```
[eleves]
ReadOnly = 0
Servername = localhost
Port = 5432
Database = eleves
Driver = /usr/lib/libodbcpsql.so
Username = mlx
```

## Utiliser une ressource ODBC dans Open Office

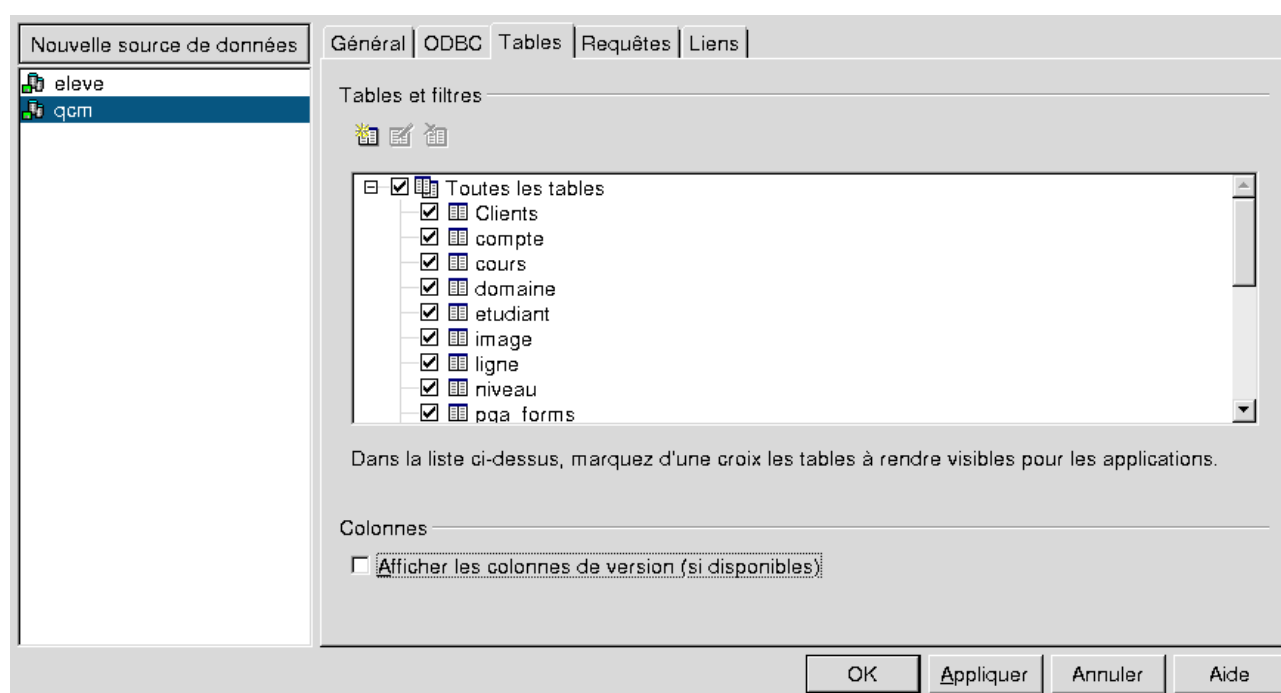
Une fois la ressource ODBC déclarée, lancer OpenOffice.org.

Allez dans le menu « Outils », « Sources de données »

Type de base de données, prenez « ODBC », enfin sélectionnez votre base « eleves » dans la liste.



Normalement c'est terminé, mais nous allons tout de suite effectuer un test. Pour cela allez sur l'onglet « tables »



Si les tables s'affichent, alors la définition de la ressource est correcte. Refermez les fenêtres, vous aurez accès à vos bases de données avec la touche de fonction « F4 ».

Maintenant que la, ou les, ressources sont définies dans OpenOffice.org, il est possible des utiliser.

### Créer un formulaire simple

Nous allons utiliser deux solutions. La première s'appuie sur les assistants, la deuxième montre comment créer un formulaire manuellement.

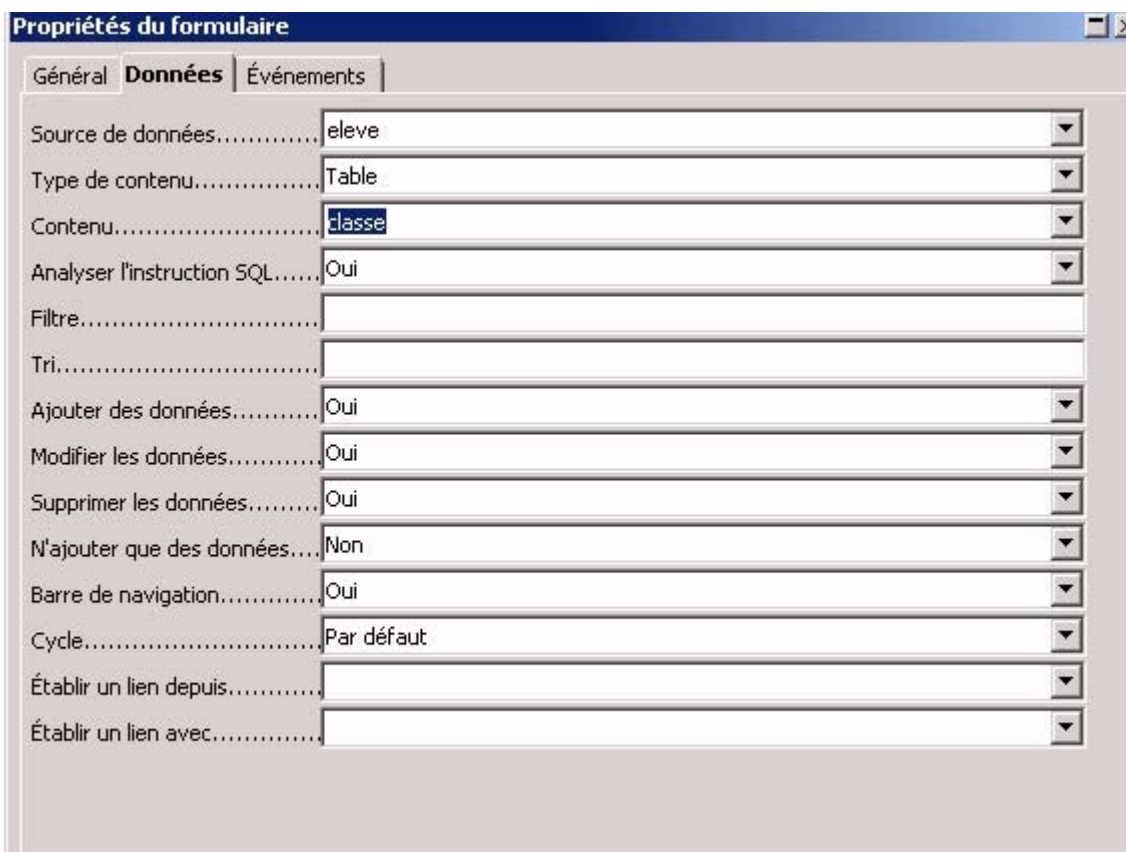
#### Première solution :

Prenez « fichier », « autopilote », « formulaire ».

Sélectionnez votre base, puis la table « élèves ». Vous pouvez utiliser une requête préalablement définie.

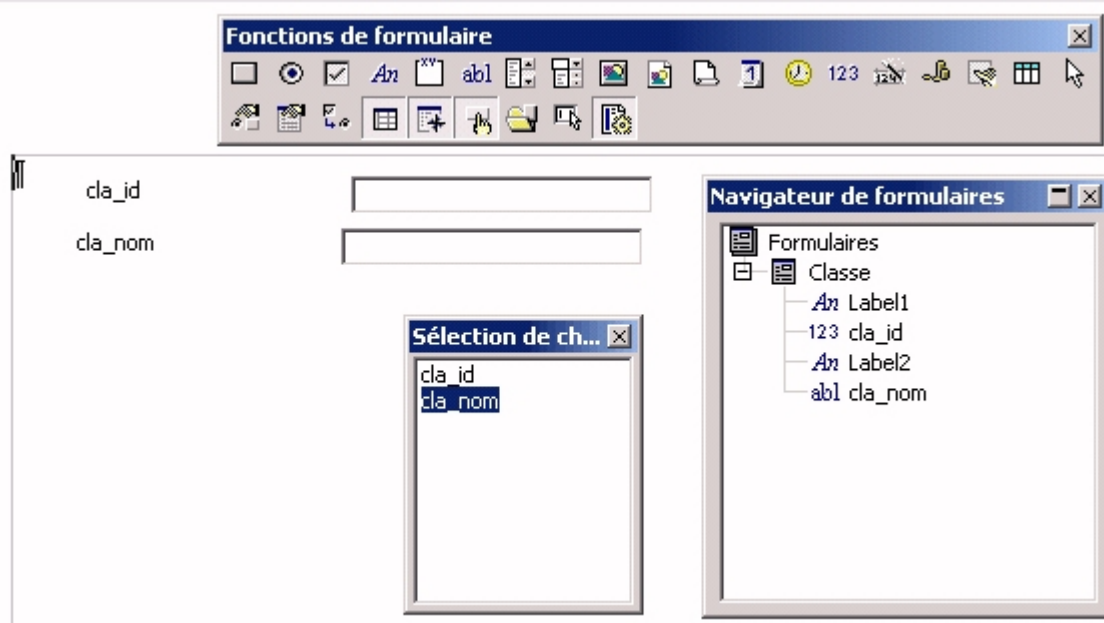
Prenez tous les champs, validez, c'est terminé.





Remplissez les champs « Source de données » et « Contenu », refermez la fenêtre.

Dans la barre d'outil activez le bouton « insérer un champ ». Une fenêtre « sélection de champ » s'active. Faites glisser les champs sur votre document.

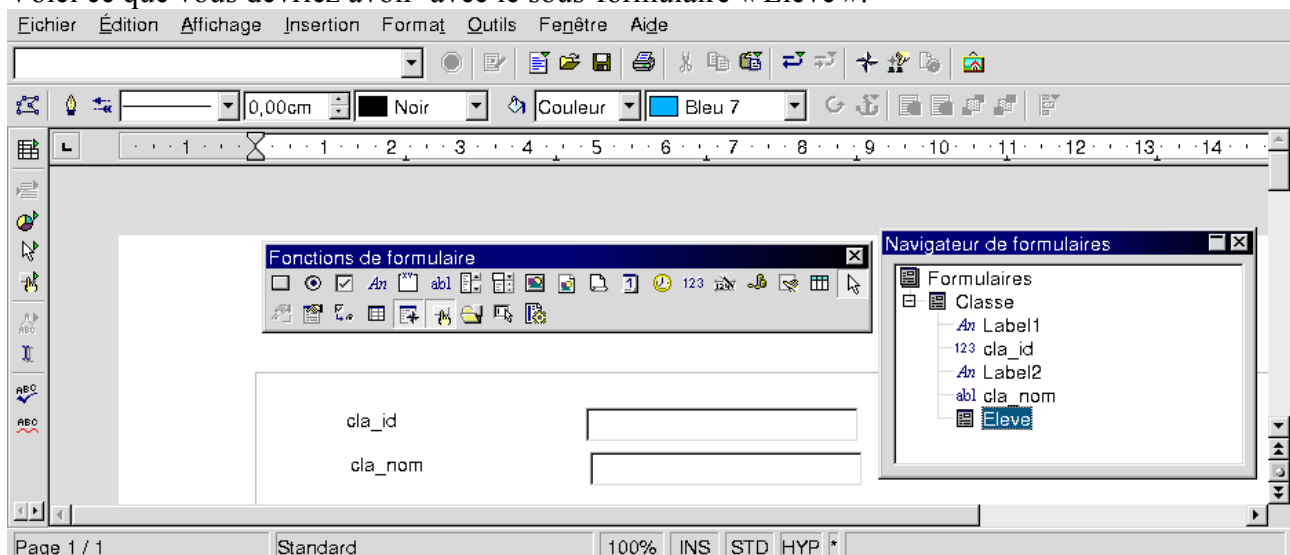


## Créer un sous formulaire avec une requête paramétrée

Ce que l'on souhaite faire maintenant est plus compliqué. On souhaite avoir par classe, la liste des élèves. Il est possible de concevoir une requête multi-tables, (classe et élèves) qui donne, par classe, la liste des élèves. Ensuite il suffit d'utiliser le résultat de cette requête dans un formulaire comme nous venons de le faire. Cette technique sera vue un peu plus loin.

Nous allons utiliser un autre procédé, celui des formulaires et des sous-formulaires avec des requêtes dites « paramétrée ». C'est, à ce stade de développement d'OpenOffice.org, le seul moyen de réaliser des sous-formulaires avec les fichiers dbf. Nous allons l'appliquer à PostgreSQL. Pour créer un sous-formulaire au formulaire « Classe », il suffit de faire un « Click DR » sur l'objet « classe », « nouveau », « formulaire », appelez le « Eleve ».

Voici ce que vous devriez avoir avec le sous-formulaire « Eleve »:

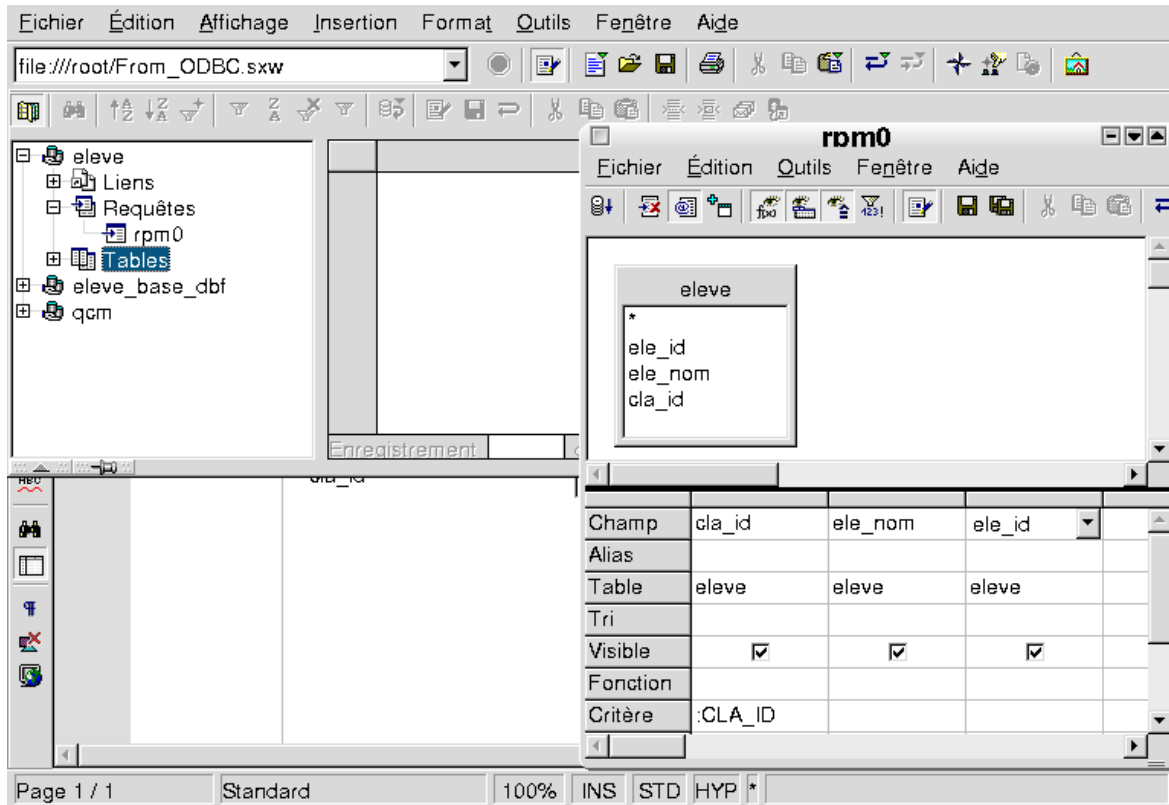


Nous allons associer une requête paramétrée à ce sous-formulaire. Cela nous permettra d'avoir les élèves pour une classe donnée. Le numéro de classe sera le paramètre.

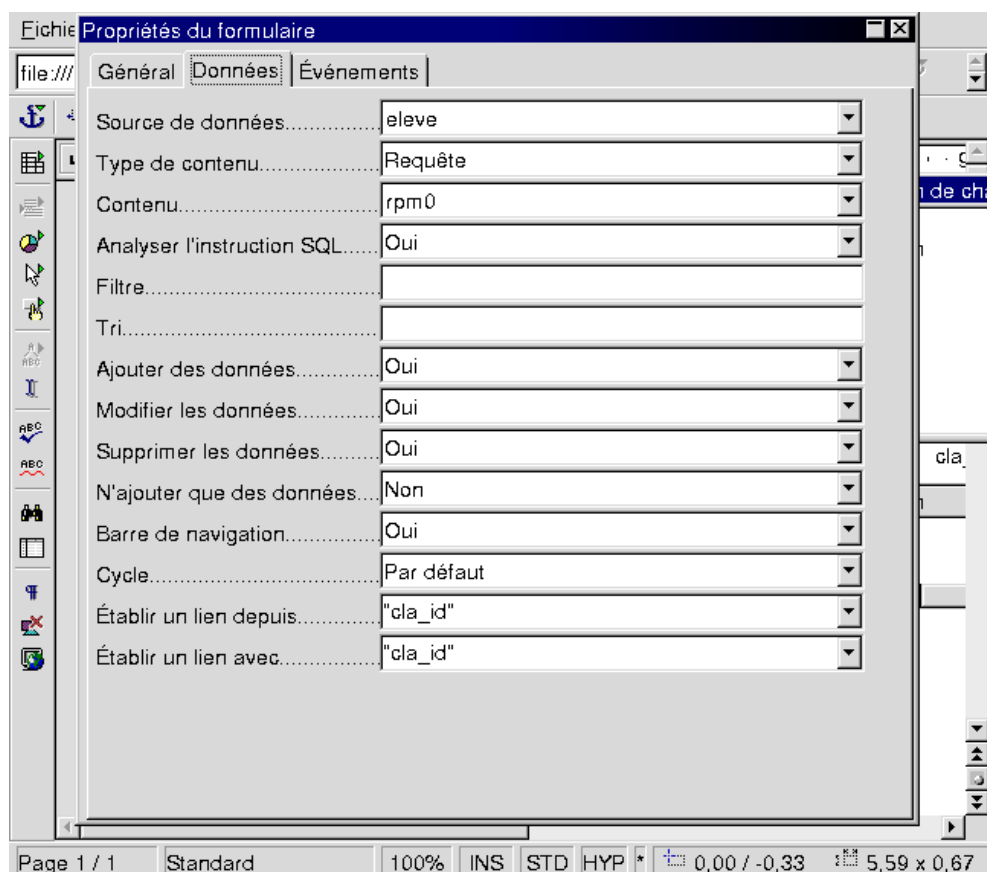
- Appuyer sur F4 pour ouvrir le menu des bases de données ou par le menu « Affichage », « Sources de données »
- Prenez votre source de données,
- Requête, puis ClickDR, « Nouvelle ébauche de requête »

Adaptez à l'aide de l'image ci-dessous, n'oubliez pas le paramètre. Vous pouvez dès maintenant tester le résultat. Quand c'est terminé, enregistrez sous le nom rpm0. J'ai mis :CLA\_ID, mais vous pouvez mettre ce que vous voulez.

Attention seulement le « : » (deux points) est important, car ils définissent une variable.



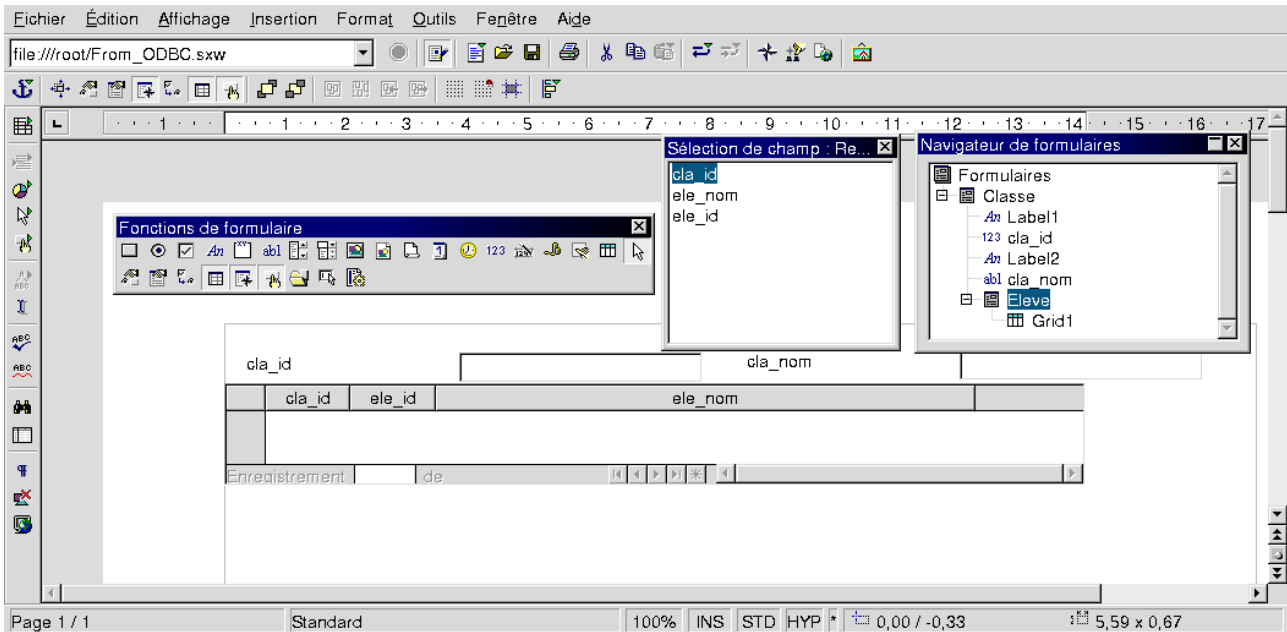
- Fermez tout et revenez au document.
- Allez dans les propriétés sur sous-formulaire « Eleve ». Remplissez les zones comme ci-dessous:



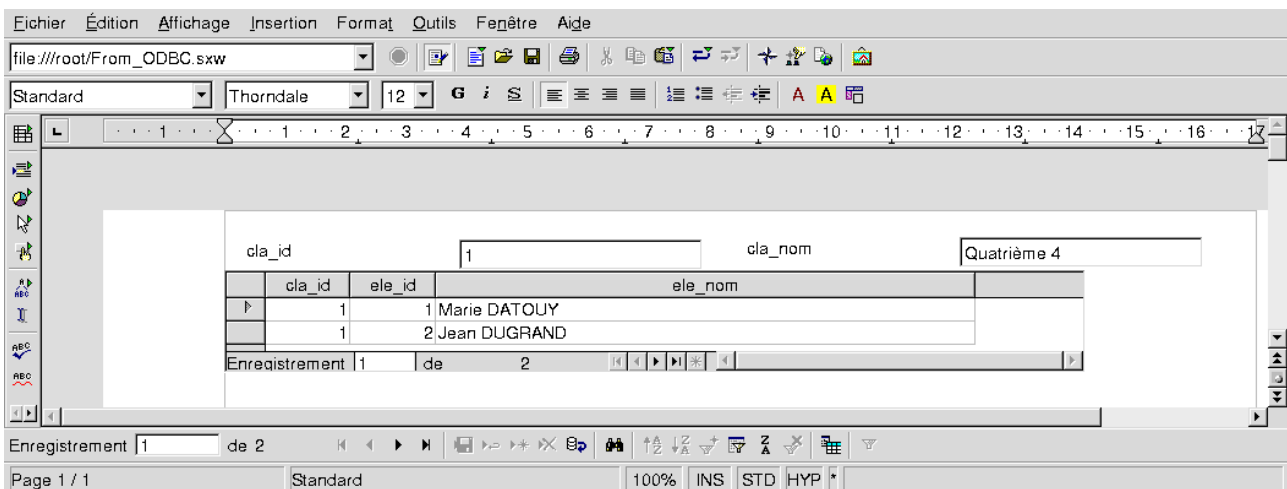
- Votre source de données
- Type de contenu « requête »
- Source, le nom de la requête paramétrée, préalablement créée
- Établir un lien depuis (Mettre le nom du champ du formulaire maître)
- Établir un lien avec (Mettre le nom du champ du sous-formulaire). C'est cette valeur qui sera passée au paramètre de la requête.

Revenez à votre document. Dans le « Navigateur de formulaire », sélectionnez bien le sous-formulaire. (Pour nous « Eleve »)

- Insérer un contrôle de table
- Sélectionner « Insérer un champ », et faites glisser les champs sur le contrôle de table



Cliquez sur le bouton de la barre d'outil « désactiver le mode conception ». Vous êtes maintenant en mode saisie/consultation de votre base de données.



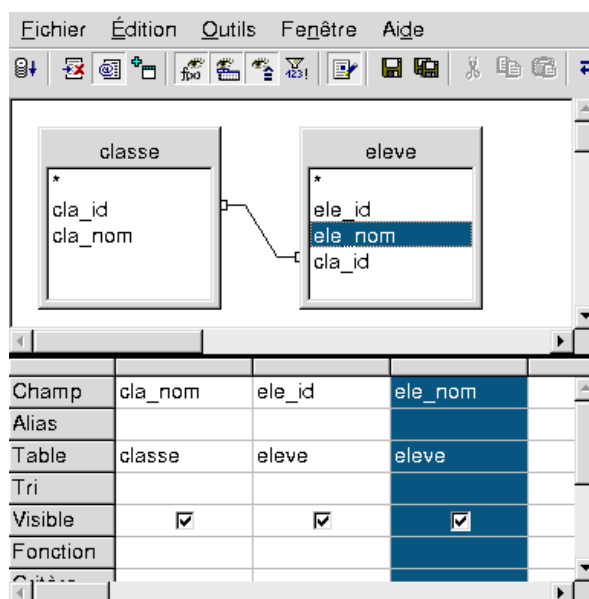
## Créer un sous formulaire avec une requête multi-tables

Les serveurs de bases de données permettent de créer des requêtes multi-tables. On va voir comment créer un formulaire qui donne pour chaque classe la liste des élèves par ce moyen.

Créez une nouvelle requête sur votre base de données :

- F4, sélectionnez votre source de données, requête, Click Dr, « nouvelle ébauche de requête ».





Enregistrez votre requête sous un nom, « rqm » par exemple.

Prenez un nouveau document, activez la barre de création de formulaire.

Créez un nouveau formulaire que vous appelez « Classe ».

Dans les « propriétés du formulaire », onglet « données », mettez :

- Source de données: eleve
- Type de données: requête
- Contenu « rqm »

Refermez la fenêtre de configuration des propriétés du formulaire et revenez à votre formulaire.

Insérez un contrôle de type « zone de texte »

Sélectionnez le contrôle, Click Dr, « Contrôle »

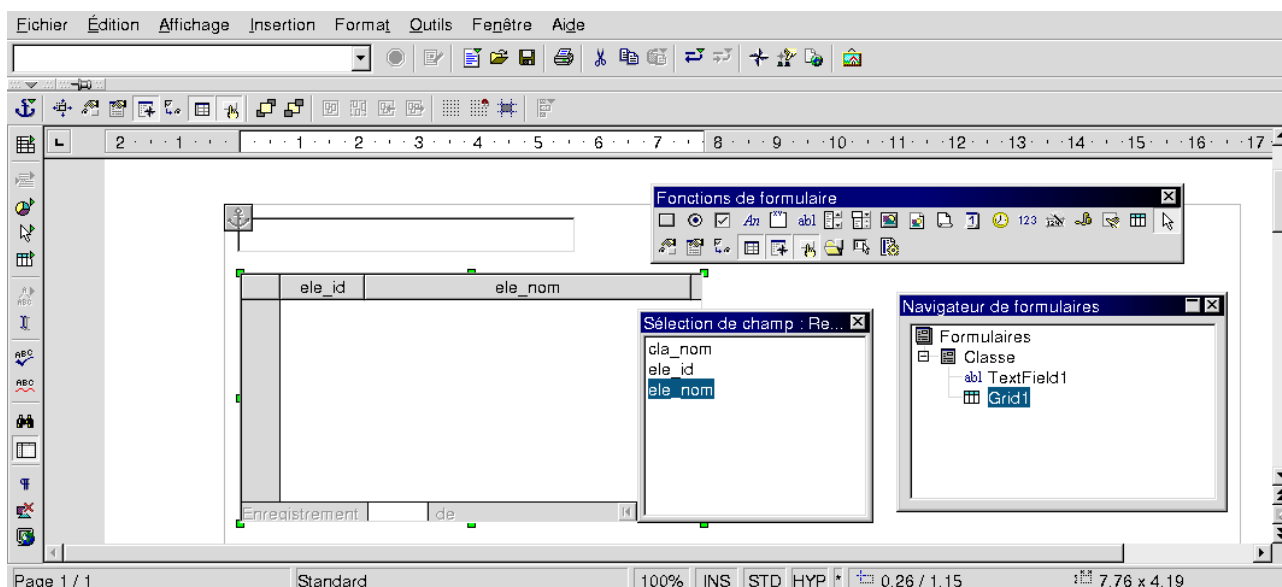
Prenez l'onglet « Données », et dans « Champ de données » mettez cla\_nom (nom de la classe).

Insérer ensuite un « Contrôle de table »

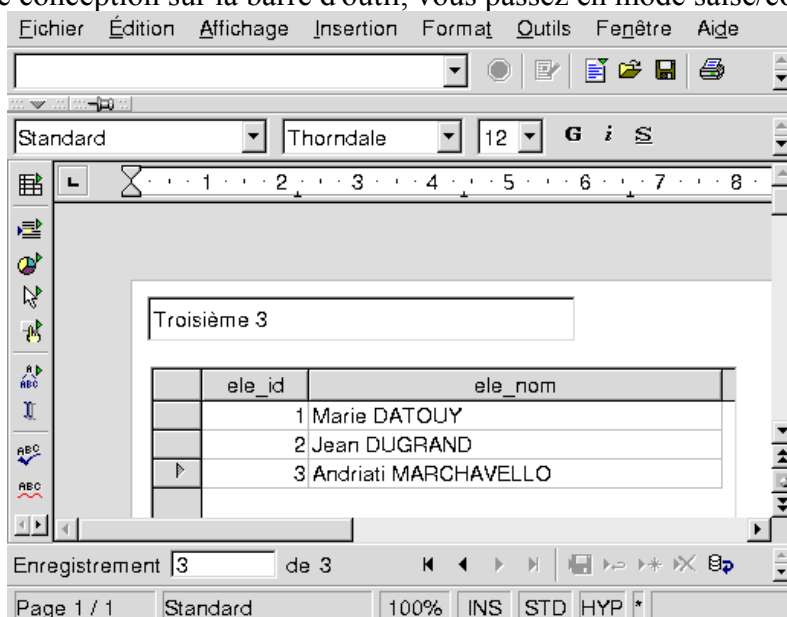
Sélectionnez le « Contrôle de table »

Prenez « Insérer un champ » dans la barre d'outil du formulaire. Faites glisser « ele\_id » et « ele\_nom », sur le contrôle de table.

Voici ce que vous devez obtenir en mode conception :



Désactivez le mode conception sur la barre d'outil, vous passez en mode saisie/consultation.



Notes sur les difficultés rencontrées :

- 1 - Vous pouvez avoir des problèmes d'identification de variables avec la requête paramétrée. Dans ce cas passez tout en minuscule (:cla\_id, dans la requête paramétrée, et cla\_id dans les propriétés « Établir un lien ... » du sous formulaire « Eleve »).
- 2 - Le changement de valeur n'est pas toujours pris en compte dynamiquement, utiliser la fonction « Tout actualiser » du menu « Outil ».

3 - Il existe un « bug » (Entre Guillemets) connu, sur l'utilisation de requêtes paramétrées ou requêtes SQL avec les bases ODBC et l'utilisation de variables déclarées. Ce dysfonctionnement dispose déjà d'un correctif mais pas encore en mode graphique (À ce jour le 8/10/2002). Le Support d'OpenOffice.org m'a envoyé un script que je mets ci-dessous. Il fonctionne parfaitement sur Linux, je n'y suis pas arrivé sous Windows, mais en fait je n'ai pas trop cherché.

Pour l'utiliser c'est simple :

Vous prenez le menu « Outils », « Macro », « Main », « Éditer », vous collez le script dedans.

Vous remplacez le nom de la base « eleve » par le nom de votre base dans la fonction « Main » :

```
Sub Main
```

```
    EnableParameterNameSubstitution( "eleve" )
```

```
End Sub
```

Vous faites « Exécuter », c'est terminé.

==== Réponse du support Open Office.org =====

*The Problem*

*When working with parametrized statements, OpenOffice.org usually uses named parameters such as in "SELECT \* FROM <table> WHERE <field> = :param." Here ":param" is a named parameter. However, some databases do not allow such named parameters, but only unnamed ones. The statement then would be "SELECT \* FROM <table> WHERE <field> = ?". Such databases usually reject statements with named parameters.*

*The Solution*

*The OpenOffice.org ODBC-SDBC bridge (more sloppy: OOo's ODBC driver) features substituting named parameters with unnamed ones before sending statements to the system driver.*

*Parameter name substitution can be enabled on a per-data-source basis. For this, the "Info" property of a data source should contain a name-value-pair with*

*Name: ParameterNameSubstitution*

*Value: TRUE*

*Unfortunately, there is no user interface, yet, for doing so. There is an issuezilla bug requesting this UI, but as long as it's not fixed, you could use the Basic macro provided below, which adds the setting for a data source of your choice.*

==== Couper ICI =====

```
REM ***** BASIC *****
```

```
Option Explicit
```

```
Sub Main
```

```
    EnableParameterNameSubstitution( "eleve" )
```

```
End Sub
```

```
Sub EnableParameterNameSubstitution( sDataSourceName as String )
```

```
    ' the data source context (ehm - the service name is historical :)
```

```
    Dim aContext as Object
```

```
    aContext = createUnoService( "com.sun.star.sdb.DatabaseContext" )
```

```
    ' the data source
```

```
    Dim aDataSource as Object
```

```
    aDataSource = aContext.getByname( sDataSourceName )
```

```
    ' append the new ParameterNameSubstitution flag
```

```
    Dim bFlag as Boolean
```

```
    bFlag = TRUE
```

```
Dim aInfo as Variant
aInfo = aDataSource.Info
aInfo = AddInfo( aInfo, "ParameterNameSubstitution", bFlag )

' and write back
aDataSource.Info = aInfo
' flush (not really necessary, but to be on the safe side :)
aDataSource.flush
End Sub

Function AddInfo( aOldInfo() as new com.sun.star.beans.PropertyValue, sSettingsName as String,
aSettingsValue as Variant ) as Variant
    Dim nLower as Integer
    Dim nUpper as Integer
    nLower = LBound( aOldInfo() )
    nUpper = UBound( aOldInfo )

    ' look if the setting is already present
    Dim bNeedAdd as Boolean
    bNeedAdd = TRUE

    Dim i As Integer
    For i = nLower To nUpper
        If ( aOldInfo( i ).Name = sSettingsName ) Then
            aOldInfo( i ).Value = aSettingsValue
            bNeedAdd = FALSE
        End If
    Next i

    ' allocate the new array
    Dim nNewSize as Integer
    nNewSize = ( nUpper - nLower + 1 )
    If bNeedAdd Then nNewSize = nNewSize + 1
    Dim aNewInfo( nNewSize ) as new com.sun.star.beans.PropertyValue

    ' copy the elements (a simply copy does not work in Basic)
    For i = nLower To nUpper
        aNewInfo( i ) = aOldInfo( i )
    Next i

    ' append the new setting, if necessary
    if ( bNeedAdd ) Then
        aNewInfo( nUpper + 1 ).Name = sSettingsName
        aNewInfo( nUpper + 1 ).Value = aSettingsValue
    End If

    AddInfo = aNewInfo()
End Function
===== Couper ICI =====
```

### **Notes complémentaires :**

Si vous désirez passer de MS Access vers PostgreSQL/Windows, il vous faut télécharger et installer PostgreSQL sur votre machine ( environnement windows) et le pilote ODBC dont on a déjà parlé. Le reste du document est utilisable.

Si vous désirez passer de MS Access vers MySQL, il vous faut télécharger et installer les pilotes ODBC pour MySQL. Ils sont sur <http://www.mysql.com>. Il n'y a pas à ma connaissance de solution libre qui permettent de transférer automatiquement une base MS Access vers MySQL. Vous pouvez procéder de la façon suivante :

- Déclarez la ressource ODBC vers votre base MySQL
- Ouvrez la base MS Access
- Pour chaque table, utilisez la fonction « exporter » du menu et utilisez le lien vers la ressource ODBC que vous venez de déclarer. La structure et les données seront exportées.

Il y a un très bon document pour MySQL sur : <http://pbs.linux.net.pl/ksiazki/OOoMySQL.pdf>  
Et d'autres excellentes sources d'informations sur : <http://www.unixodbc.org/>

## **Migrer une base de données MS Access vers un format DBF**

L'avantage du format .dbf est que, vous pouvez directement depuis OpenOffice.org créer vos bases et vos nouvelles tables. Nous allons voir comment récupérer un existant provenant de MS Access.

Ouvrez votre base de données access. Pour chaque table utilisez la fonction « exporter » (Click DR). Prenez un format Dbase dans « Type de fichier », enregistrez vos bases dans un répertoire.

Note : J'ai essayé Dbase 3, Dbase 4 et Dbase 5, tous les formats ont l'air de fonctionner correctement.

## **Utiliser un format Dbase sous OpenOffice.org**

1 - Il faut créer un répertoire pour stocker les tables de votre base. On crée un répertoire « eleves » pour stocker les tables de la base élèves.

Sous Windows, ça donne sur ma machine : « C:\Program Files\OO\user\database\eleves »

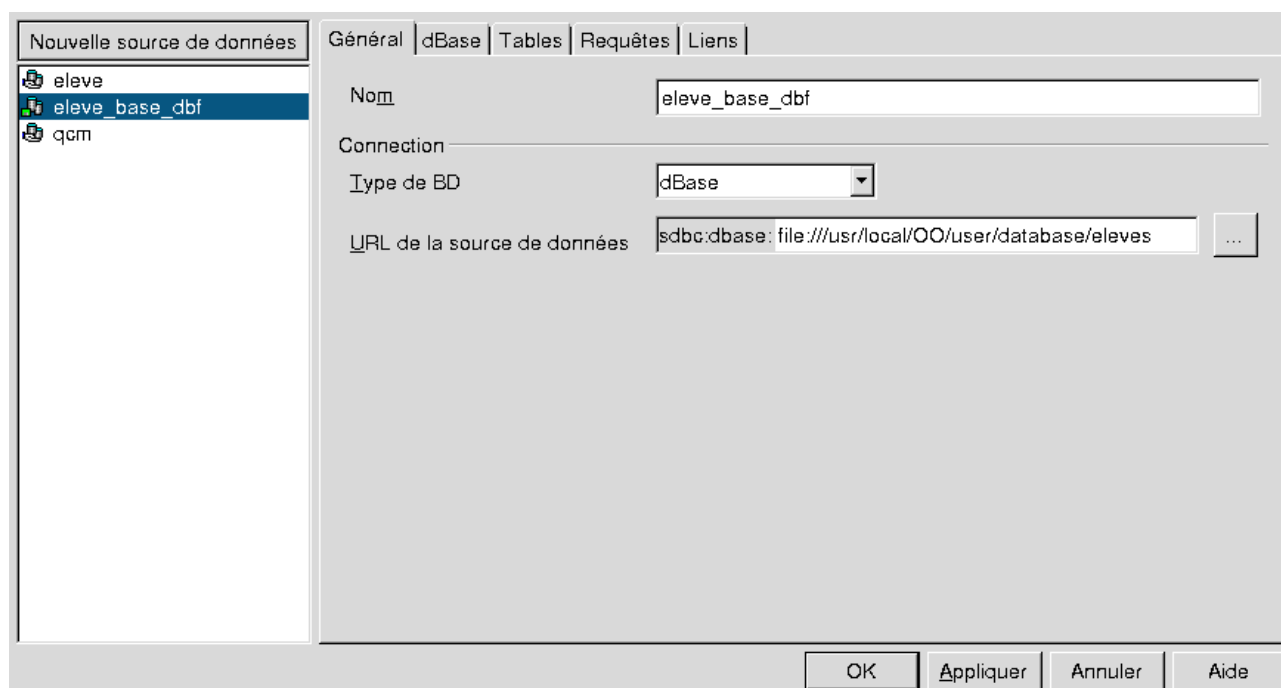
Sous Linux, ça donne, toujours sur ma machine : « /usr/local/OO/user/database/eleves »

2 - Ensuite vous copiez les fichiers dbf auparavant enregistrés dans le répertoire créé. Si vous avez un partage samba, cela se fait sans difficulté.

3 - Dans OpenOffice.org, on va définir la source de données.

Prenez « nouvelle source de données », donnez lui un nom.

Type de BD, prenez Dbase, est sélectionnez le répertoire dans lequel vous avez stocker les fichiers DBF.



C'est terminé, on va utiliser cette source de données pour créer un formulaire et un sous-formulaire.

Notes :

Sous Linux, les tables doivent apparaître sur l'onglet « tables ». Si ce n'est pas le cas et si vous n'avez pas commis d'erreur de configuration, vérifiez que les noms des tables ne sont pas en MAJUSCULE. Si oui, renommez les. Exemple : « ELEVE.DBF » devient « eleve.dbf ».

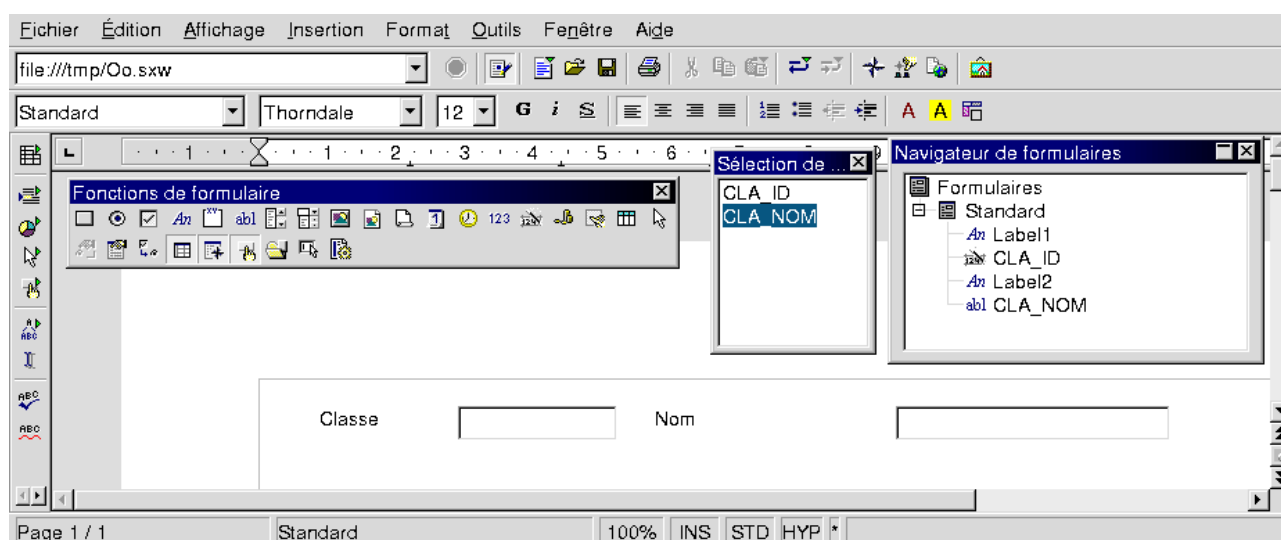
Sous Windows ou sous Linux, vous pouvez avoir des difficultés d'affichage avec les caractères accentués. Une fois la source de données définie, dans OpenOffice.org, prenez l'onglet « dbase » et dans jeu de caractères, prenez « IBMPC (850) / DOS ».

## Créer un formulaire et un sous formulaire avec des fichiers DBF

Le principal problème avec les .dbf est que l'on ne peut pas faire de requêtes multi-tables. Pour avoir dans un formulaire « les classes », puis pour chaque classe « les élèves », on utilisera un sous-formulaire. Voici de façon détaillée comment faire (aidez vous du paragraphe et des images écrans sur PostgreSQL) :

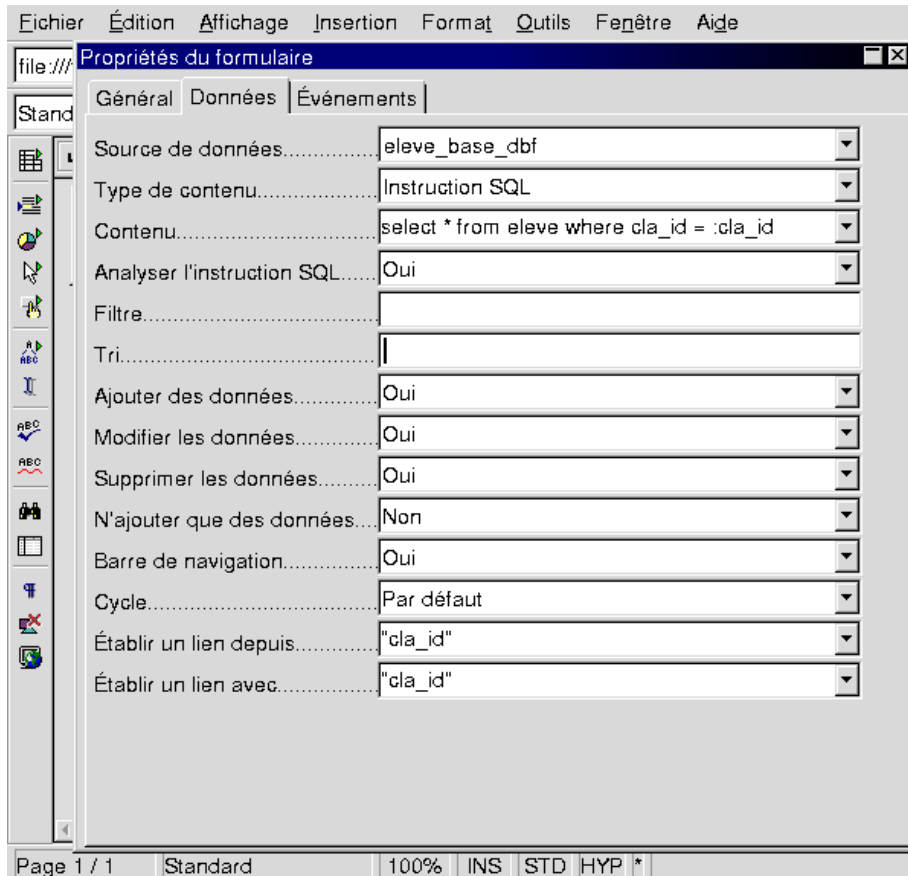
- Ouvrir un nouveau document texte, activez la barre d'outils des formulaires.
- Activer le navigateur de formulaire
- Créer un formulaire principal « classes »
- Dans « Propriétés du formulaire », mettez votre source de données DBF, et dans le champ « Données » mettez « classe »
- Sélectionnez le bouton « Insérer un champ », faites glisser les champs de la table classe dans le formulaire.

Voici ce que ça doit à peu près donner :



Il reste à insérer pour chaque classe les élèves. On va utiliser un sous-formulaire. Le problème est qu'il faut établir un lien entre le numéro de classe de la table élève et le numéro de classe de la table classe, afin que seuls les élèves d'une classe donnée apparaissent dans le sous-formulaire.

- Créez un sous-formulaire « Eleve » à la table classe. (Click DR, nouveau , formulaire...)
- Remplissez les propriétés en vous aidant de l'image ci-dessous :



- « Source de données », ça ne bouge pas, c'est toujours la même.
- « Type de contenu », vous pouvez utiliser comme ici instruction SQL, ou requête. Si vous mettez requête, il vous faut auparavant avoir créé une requête paramétrée. Cette option n'est pas vue ici.
- Dans « contenu », mettez l'instruction SQL. Ici on prend tous les champs. (select \*)
- Dans « contenu » l'expression « where cla\_id = : cla\_id » est un peu particulière.

cla\_id est le nom du champ de la table « eleve » qui contient le numéro de classe.

:cla\_id est un paramètre qui sera passé à la requête. Il reçoit la valeur d'un numéro de classe courant. Pour nous ce sera celui du formulaire « père », c'est à dire du formulaire « classe ». Vous pouvez mettre n'importe quoi comme variable :x, :y, :toto...

Ce paramètre sera renseigné par la valeur donnée dans « Établir un lien avec »

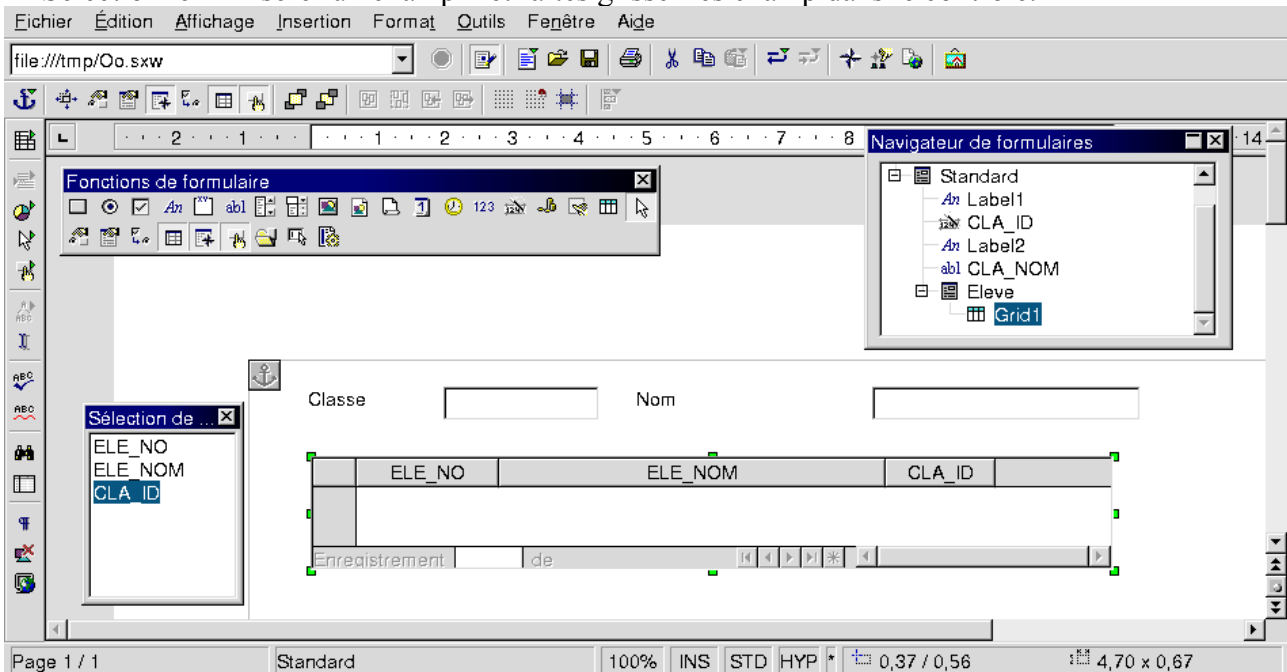
Il faut maintenant établir un lien entre le numéro de classe « cla\_id » du formulaire « Elève » et celui du formulaire « Classe ». Cela se fait avec :

- Établir un lien depuis : mettre ici le nom du champ du formulaire maître (parent)
- Établir un lien avec : mettre ici le nom du champ du formulaire dépendant (enfant)

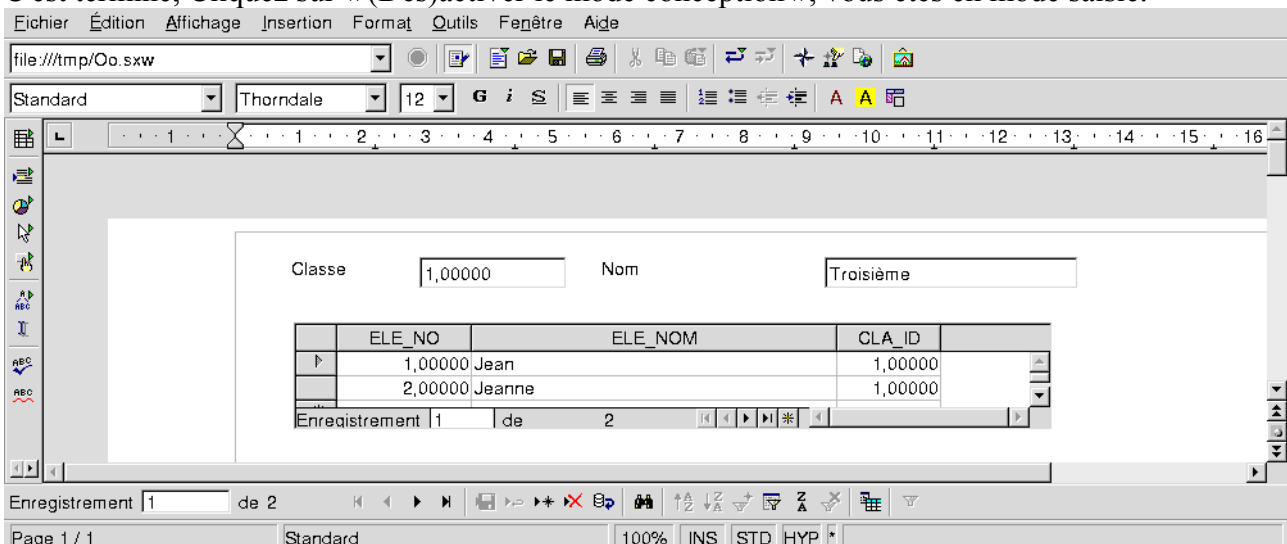


On dispose d'une relation de type 1 ---> N. Pour une classe dans le formulaire, nous allons avoir la représentation de N élèves. Il faut que l'on mette les champs dans le sous formulaire pour les élèves. Il y a plusieurs solutions, on va utiliser une représentation tabulaire à l'aide du champ « Contrôle de table ».

- Sélectionnez dans le « Navigateur de formulaire » le formulaire « Eleve ».
- Insérez un champ « Contrôle de table » sur votre document.
- Sélectionner « Insérer un champ » et faites glisser les champ dans le contrôle.



C'est terminé, Cliquez sur « (Dés)activer le mode conception », vous êtes en mode saisie.



Note: Si vous rencontrez un problème à ce niveau là, revenez dans la définition des paramètres du sous-formulaire, passez tout en majuscule. (cla\_id, devient CLA\_ID, dans l'exemple que je vous ai

donné.)

### **Fin de notre itinéraire :**

Voilà, ici on se sépare. Je n'ai plus qu'à vous souhaiter bonne continuation. J'espère que cette petite introduction pourra en aider quelque uns. Pour ma part j'ai découvert cet aspect d'OpenOffice.org et j'ai été assez agréablement impressionné.

Autres sources d'information sur OO en Français :

<http://fr.openoffice.org/>

<http://fr.openoffice.org/index-navigation.html>

<http://whiteboard.openoffice.org/lingucomponent/>

<http://www.cspi.qc.ca/recit/formation/staroffice.htm>

*Alix MASCRET*

## Crédits

**Auteur** : Alix MASCRET

**Dernière modification** : 14 mai 2002

**Contacts**: OpenOffice.org Documentation <http://fr.openoffice.org>

## 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 OpenOffice.org et les bases de données. The Initial Writer of the Original Documentation is Alix Mascret Copyright (C) 2002. All Rights Reserved. (Initial Writer contact(s): [mascret@beaupeyrat.com](mailto:mascret@beaupeyrat.com)).

Contributor(s): \_\_\_\_\_.  
Portions created by \_\_\_\_\_ are Copyright (C) \_\_\_\_\_ *[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.