

How to « Lier des tables à l'aide de paramètres nommés SQL »

Diffusé par

Le Projet Documentation OpenOffice.org

Table des Matières

Introduction.....	3
Créer la base de donnée, les tables, et la source de données.....	5
Ajouter des données aux tables.....	9
Permettre l'usage des paramètres nommés pour la source de données.....	11
La Macro	12
Créer un formulaire pour une source de données.....	15
Ajouter un sous-formulaire.....	18
Lier un sous-formulaire à un formulaire principal.....	23
Formatage des champs de données.....	27
Quelques exemples supplémentaires.....	31
Crédits	36
Licence.....	36

Introduction

Ce document tente de montrer comment créer un formulaire maître et un formulaire détail lié pour une base de données MySQL via ODBC. Vous pourriez ainsi vouloir afficher un auteur et tous les livres qu'il a écrit, ou un acteur et tous les films dans lesquels il est apparu, ou bien encore un client et toutes les commandes placées par celui-ci.

Deux tables d'une base de données sont liées à l'aide d'une requête SQL contenant un paramètre nommé (voire plus ultérieurement). L'exemple que vous allez construire utilise les clients d'une table **client** et les commandes de ces mêmes clients reprises dans une autre table **commandes**. Chaque commande est liée au client qui l'a placée au moyen de la valeur *cl_id* dans la table **commandes** qui doit correspondre à la valeur *client_id* du client correspondant dans la table **clients**.

Il est par ailleurs utile de rappeler la différence qu'il existe entre les tables de la base de données MySQL elle-même, qui contiennent les données et qui sont gérées par le serveur MySQL, et les outils graphiques OOo qui affichent les données issues des tables. Vous allez utiliser ces outils, comme les champs et les grilles de données, pour créer un ensemble de formulaires liés.

L'objectif sera de créer un formulaire avec des champs libellés pour afficher les données d'un client particulier et un sous-formulaire contenant une grille représentant toutes les commandes pour ce client. Quand les deux formulaires seront liés ensemble de manière adéquate, vous serez capable de naviguer à travers la liste de clients, un à un, et ainsi avoir la grille contenant les commandes mise à jour dynamiquement.

Cet exemple a été construit en utilisant une version mono-utilisateur et autonome de OOo 1.0.1 sur Windows98. Le modus operandi devrait pouvoir être appliqué aux installations multi-utilisateurs et sur d'autres OS, mais cela n'a pas été testé.

MySQL est disponible à l'adresse <http://www.mysql.com/> .

Nous partirons du principe que vous avez déjà installé MySQL et, que vous pouvez créer et publier des bases de données MySQL, que vous savez configurer une source de données ODBC dans OOo et faire apparaître, dans OOo, les données issues des tables, ainsi que vous êtes capable d'ajouter, d'effacer, et de modifier des enregistrements. Souvenez-vous que OOo ne peut éditer que des tables MySQL qui ont une clé primaire ou une colonne index unique. Les autres tables sont en lecture seule.

Si vous avez besoin d'aide pour la création de la source de données nécessaire à cet exemple, jetez un coup d'oeil au How to "Création d'une base de données Ab Initio" à l'adresse suivante <http://fr.openoffice.org/servlets/ProjectDocumentList>

Pour de l'aide à propos de ODBC, vous pouvez consulter le document "OpenOffice.org 1.0, unixODBC, and MySQL" à <http://fr.openoffice.org/servlets/ProjectDocumentList?dcID=724&action=download>

De nombreuses informations sont contenues dans ce documents, vous pouvez tirer avantage à le lire avant tout.

Le procédé devrait être sensiblement identique pour des bases de données comme PostgreSQL, Oracle, Firebird (Interbase), MS SQL Server, ainsi que d'autres SGBDR qui peuvent utiliser ODBC. Notez que des tables dBase peuvent être liées ensemble mais le résultat est incertain.

Enfin, le même procédé devrait également fonctionner si la connexion est faite via JDBC. Voyez à ce sujet le How to "Comment connecter une base de données MySQL via un driver JDBC" à l'adresse <http://fr.openoffice.org/servlets/ProjectDocumentList>

Créer la base de donnée, les tables, et la source de données

Créez d'abord la base de données. Nous utilisons ici MySQL v3.23.49 et myodbc v2.50.37.

```
Create database commandes_clients;

Create table clients (
    client_id int unsigned not null primary key,
    nom_client char(50));

Create table commandes (
    commande_id int unsigned not null primary key,
    cl_id int unsigned references clients.client_id,
    commande_objet char(50));
```

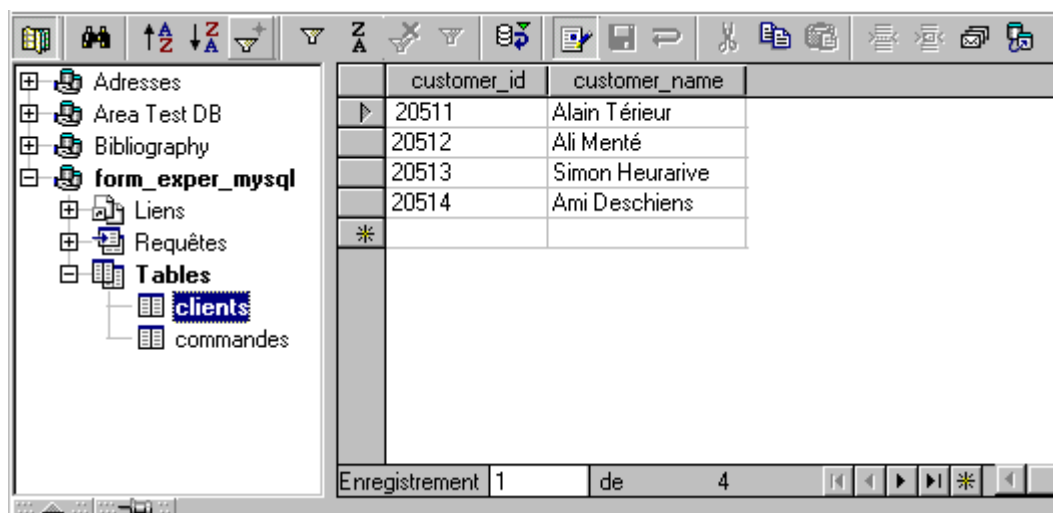
Créez ensuite la source de données ODBC. Voyez pour cela le How to “Comment créer une base de données Ab Initio” mentionné plus haut.

Dans notre exemple, nous nommons la source de données “form_exper_mysql”.

(Les pages qui suivent sont un exercice à la création et à la modification des structures de tables dans OOO. Si vos tables sont déjà conformes à ce que vous désirez qu'elles soient, vous pouvez passer à la section Ajouter des données aux tables.)

Si vous avez procédé à la création de la source de données ODBC vous pouvez créer les tables directement depuis l'**Explorateur de Sources de Données** OOo, panneau de gauche sur l'illustration (F4 ou **Affichage > Sources de Données**).

À la droite se trouve le « **Beamer** » ou éditeur de source de données.

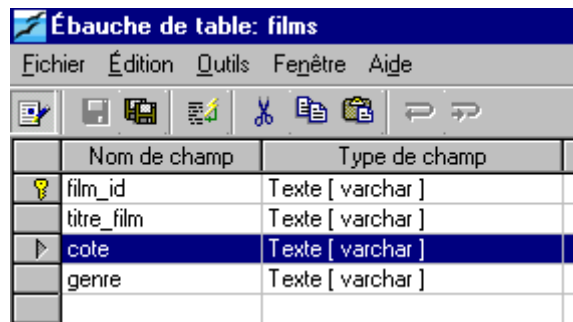


Vous pouvez en apprendre plus sur l'explorateur de sources de données ainsi que le Beamer dans l'aide OOo **Aide > Sommaire > Onglet Index** et en tapant dans le terme recherché les mots **source de données**. Lisez l'article **Source de Données; Vue d'ensemble** (ou **Explorateur de Source de Données; Vue d'ensemble**).

Pour créer une nouvelle table, cliquez sur le mot **Tables** en dessous de la source de données dans l'explorateur de sources de données, cliquez ensuite avec le bouton droit de la souris et choisissez **Nouvelle ébauche de table**.

Vous pouvez en savoir plus sur la création de tables dans l'aide OOo **Aide > Sommaire > Onglet Index** en tapant le terme recherché **ébauche de table**. Regardez l'article **ébauche de table; source de données**. Consultez également le How-to « **OpenOffice.org 1.0, ODBC, et MySQL** » mentionné plus haut.

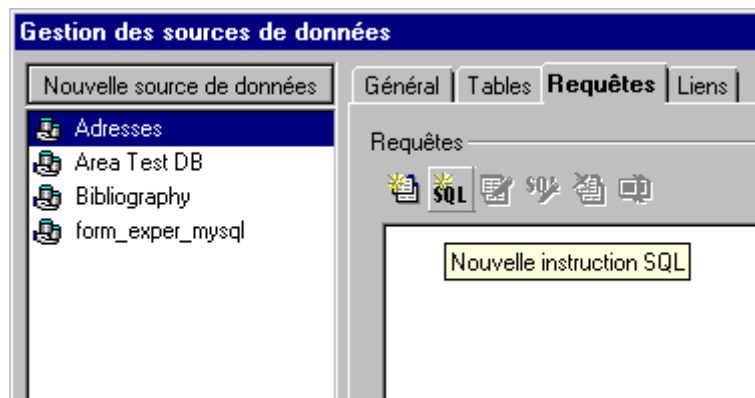
La fenêtre d'ébauche de table semble ne permettre qu'un certain type de changements sur la structure d'une table déjà créée. Par exemple, vous pouvez ajouter et effacer des champs (colonnes). Dans l'illustration ci-dessous, nous avons ajouté deux nouveaux champs (**cote** et **genre**) à une table non liée appelée films. (La table films ne fait pas partie de notre exemple. On l'utilise ici afin d'éviter d'effectuer de trop nombreux changements sur les tables clients et commandes.)



	Nom de champ	Type de champ
🔑	film_id	Texte [varchar]
	titre_film	Texte [varchar]
▶	cote	Texte [varchar]
	genre	Texte [varchar]

Vous ne pourrez cependant pas utiliser l'outil ébauche de table afin de modifier la taille d'une colonne existante. Les réglages des colonnes existantes apparaissent grisés. Il semble que ce soit une fonctionnalité manquante dans le driver ODBC mais qui pourrait être ajoutée dans les prochaines révisions.

Vous pouvez néanmoins opérer un changement de taille d'une colonne hors de OOO et travailler directement en MySQL. Cependant, il existe un moyen d'exécuter des commandes SQL directement depuis OOO. Ouvrez la fenêtre d'Administration de Sources de Données en choisissant **Outils > Sources de Données**. Sélectionnez votre source de données, sélectionnez l'onglet **Requêtes** et cliquez sur le bouton **Nouvelle instruction SQL**.



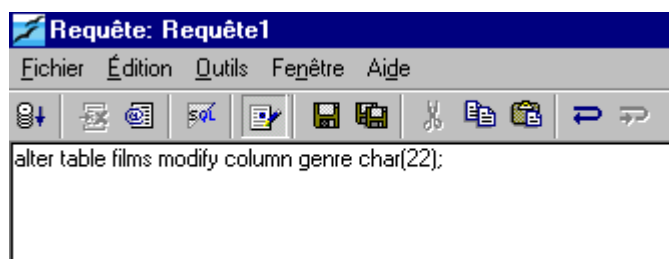
Vous obtiendrez ensuite (illustration suivante) une fenêtre Requête dans laquelle vous pouvez saisir votre instruction SQL.

Faisons un essai sur la table films.

Dans l'illustration suivante, nous avons introduit une instruction afin de changer la taille de la colonne genre dans la table films de char(16) à char(22).

```
Alter table films modify column genre char(22);
```

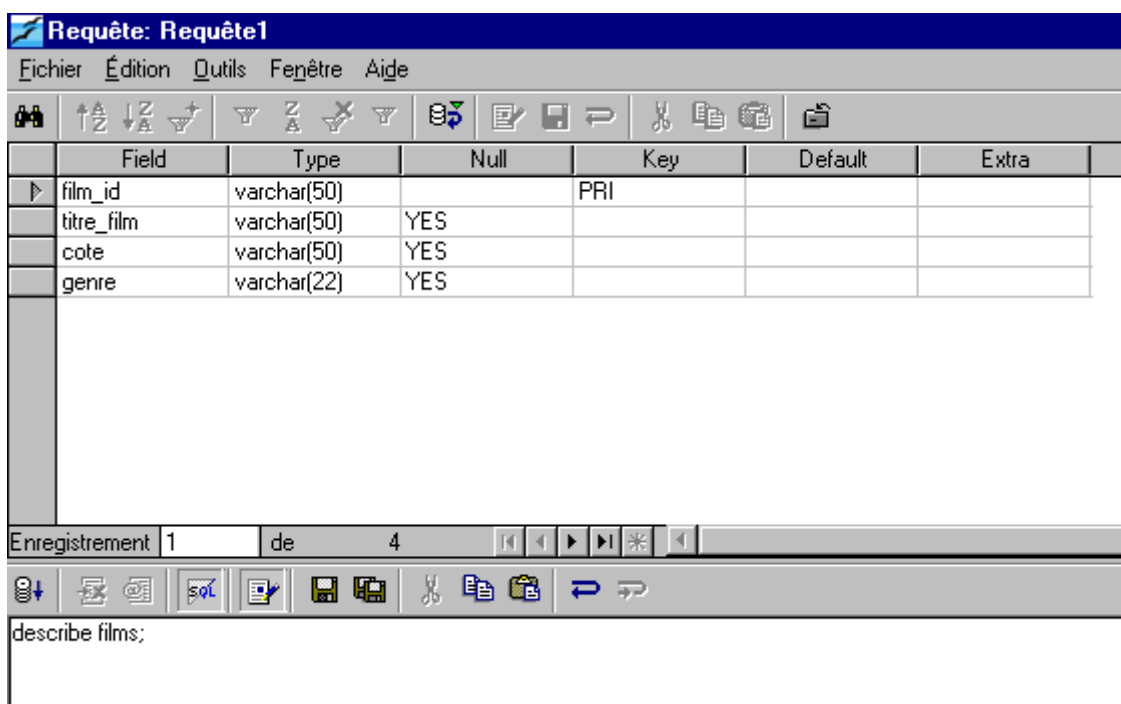
Appuyez ensuite sur le bouton Exécuter directement instruction SQL (quatrième bouton à partir de la gauche) et pressez ensuite le bouton Exécuter (le bouton le plus à gauche) pour exécuter l'instruction SQL.



Quand c'est fait, introduisez une autre commande

```
describe films;
```

et appuyez ensuite sur le bouton Exécuter. Vous devriez voir maintenant un listing des champs de la table films ainsi que la nouvelle taille du champ genre.



Vous pouvez en savoir plus à propos des commandes SQL directes dans l'aide OOO [Aide > Sommaire > Onglet Index](#) et en tapant dans le terme recherché [SQL](#). Consultez l'article [SQL; Exécution directe](#) et [Instruction SQL; Exécution](#).

Ajouter des données aux tables

Revenons maintenant à nos tables clients et commandes. Si tout a fonctionné correctement, vous devriez être en mesure de voir les deux noms des tables dans l'Explorateur de bases de données ainsi que 2 grilles vides avec des colonnes libellées dans l'explorateur de table.

Vous allez maintenant introduire quelques données dans les deux tables.

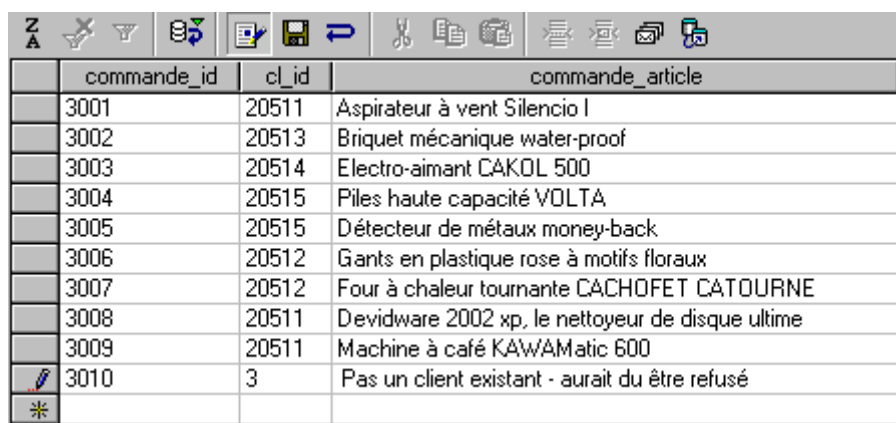
Table clients:

	client_id	nom_client
▶	20511	Alain Térieur
	20512	Ali Menté
	20513	Simon Heurariw
	20514	Ami Deschiens
*		

Table commandes

	commande_id	cl_id	commande_article
▶	3001	20511	Aspirateur à vent Silencio I
	3002	20513	Briquet mécanique water-proof
	3003	20514	Electro-aimant CAKDL 500
	3004	20515	Piles haute capacité VOLTA
	3005	20515	Détecteur de métaux money-back
	3006	20512	Gants en plastique rose à motifs floraux
	3007	20512	Four à chaleur tournante CACHOFET CATOURNE
	3008	20511	Devidware 2002 xp, le nettoyeur de disque ultime
	3009	20511	Machine à café KAWAMatic 600
*			

MySQL ne force pas encore l'intégrité référentielle des clés étrangères, il vous incombe donc de bien vous assurer que la valeur contenue dans le champ `cl_id` de la table commandes correspond à une valeur `client_id` dans la table clients.



	commande_id	cl_id	commande_article
	3001	20511	Aspirateur à vent Silencio I
	3002	20513	Briquet mécanique water-proof
	3003	20514	Electro-aimant CAKOL 500
	3004	20515	Piles haute capacité VOLTA
	3005	20515	Détecteur de métaux money-back
	3006	20512	Gants en plastique rose à motifs floraux
	3007	20512	Four à chaleur tournante CACHOFET CATOURNE
	3008	20511	Devidware 2002 xp, le nettoyeur de disque ultime
	3009	20511	Machine à café KAWAMatic 600
	3010	3	Pas un client existant - aurait du être refusé
*			

A titre d'illustration, nous avons ajouté une commande avec la la valeur **commande_id 3010** pour un client inexistant (**cl_id = 3**) . Si un contrôle d'intégrité avait été exécuté, il n'aurait pas été possible d'entrer pareille commande. Cette commande sera donc orpheline et ne sera jamais affichée dans le grille détail que nous allons construire car cette commande n'est liée à aucun des clients existants. Effacez donc cet enregistrement.

Le contrôle de l'intégrité référentielle, qui devrait être implémenté dans les versions futures de MySQL, devrait aussi éviter d'effacer un client de la table clients aussi longtemps que subsiste des commandes dans la table commandes pour ce même client.

De même que si nous changeons la valeur client_id dans la table clients, l'intégrité référentielle pourrait automatiquement mettre à jour la valeur cl_id pour chaque commande dans la table commandes afin qu'elle reflète la nouvelle valeur.

Beaucoup d'autre Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) supportent et contrôlent le mécanisme d'intégrité référentielle.

Permettre l'usage des paramètres nommés pour la source de données

Afin de permettre l'affichage d'un client et de toutes ses commandes, OpenOffice.org utilise des paramètres nommés pour lier deux tables ensemble:

```
Select * from orders where cl_id = :client_id_courant
```

Il n'y a pas d'interface utilisateur (boîte de dialogue) disponible en OOO 1.0.1 pour activer la reconnaissance des paramètres nommés. En attendant un interface utilisateur, vous devrez utiliser une **macro** pour activer la reconnaissance des paramètres nommés pour chaque source de données pour laquelle vous avez besoin de cette fonctionnalité. Cependant ce problème a été adressé; ce ne devrait donc plus être nécessaire à l'avenir.

La macro est issue de la FAQ du projet d'accès aux bases de données à l'adresse suivante:

<http://dba.openoffice.org/FAQ/> (voir question 3 de la FAQ).

La macro ainsi que les explications peuvent être trouvées à l'adresse:

http://dba.openoffice.org/howto/param_subst.html Ce document est inclus ci-dessous, avec la permission des auteurs.

La macro comporte trois parties:

```
Sub Main,  
Sub EnableParameterNameSubstitution et  
Function AddInfo
```

Depuis le menu principal de OOO, choisissez **Outils > Macro**. Dans le panneau central de la boîte de dialogue, sous **soffice > Standard**, sélectionnez Module1 et cliquez sur le bouton Éditer. Cela va ouvrir la fenêtre EDI de BASIC où vous pourrez introduire le code de la macro.

Copiez les trois parties du code de la macro depuis la section sur fond jaune ci-dessous et collez le contenu dans la fenêtre de l'EDI.

Dans la sous-routine Main, changez le nom entre guillemets de « Bibliography » vers le nom de votre source de données.

```
Sub Main  
  EnableParameterNameSubstitution( " form_exper_mysql " )  
End Sub
```

C'est la seule manipulation que vous devez exécuter sur la macro, du moins pour cette source de données.

La Macro

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

Option Explicit

' Enter the name of the data source here:

Sub Main
    EnableParameterNameSubstitution( "Bibliography" )
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

' suite à la page suivante
```

```
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

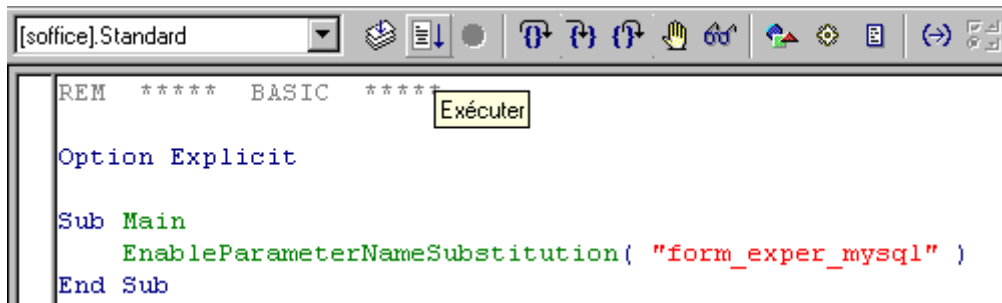
  ' copy the elements (a simply copy does not work in Basic)
  For i = nLower To nUpper
    aNewInfo( i ) = aOldInfo( i )
  Next i

  ' 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

' fin de la macro
```

Cliquez sur le bouton exécuter pour lancer la macro.



Fermez ensuite l'EDI BASIC.

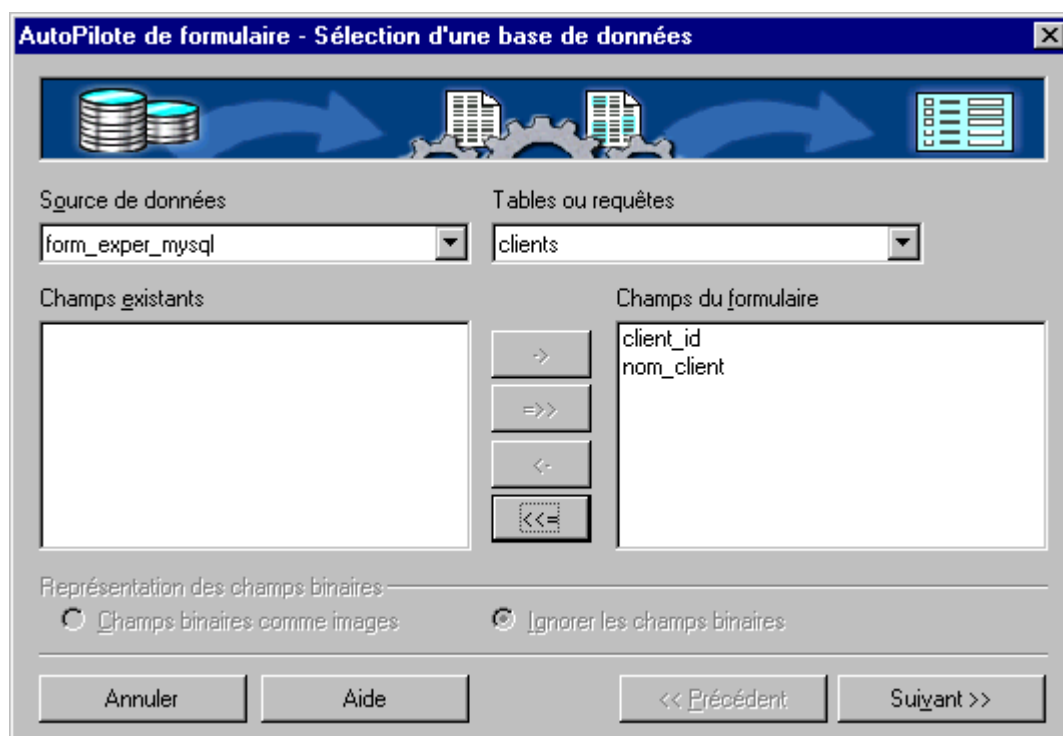
Pour rappel, le recours à cette macro est appelé à disparaître dans un futur proche.

Créer un formulaire pour une source de données

Dans le menu OOO, choisissez **Fichier > AutoPilote > Formulaire**.

Dans la première boîte de dialogue, sélectionnez la source de données et sélectionnez la table du formulaire principal (dans notre cas la table clients).

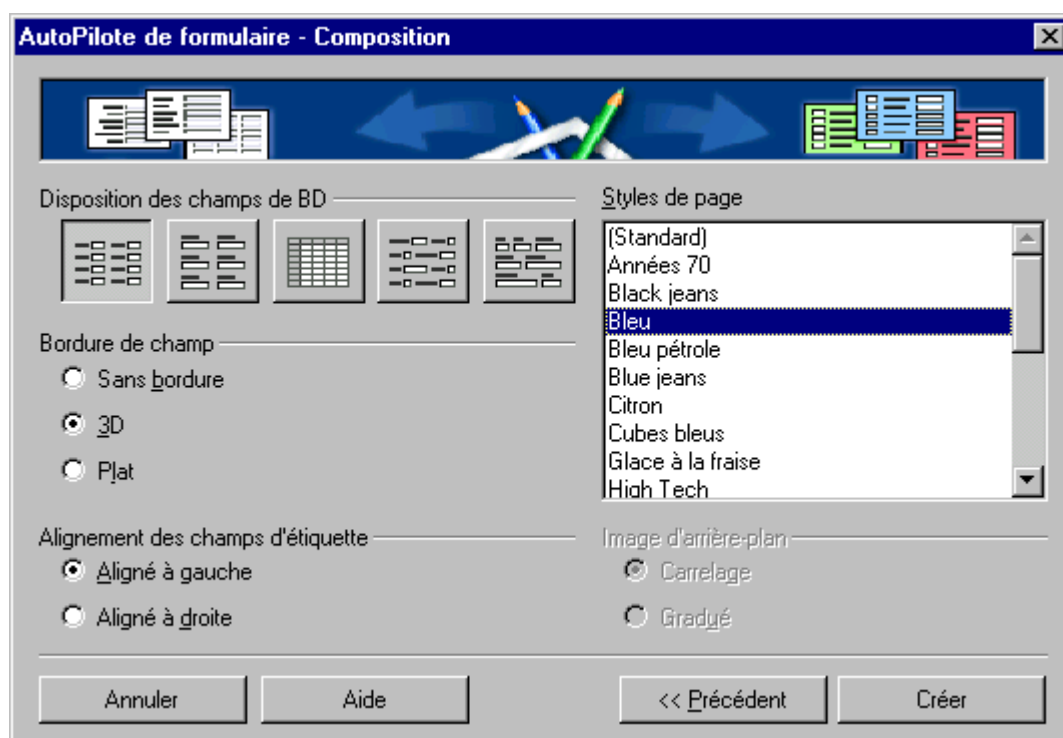
Cliquez ensuite le bouton **==>** pour inclure tous les champs (colonnes) dans le formulaire.



Cliquez maintenant sur le bouton **Suivant >>**.

Dans la seconde boîte de dialogue, cliquez sur un des cinq boutons pour choisir la disposition des champs de BD.

Sélectionnez également un style de page disponible dans la liste de droite.



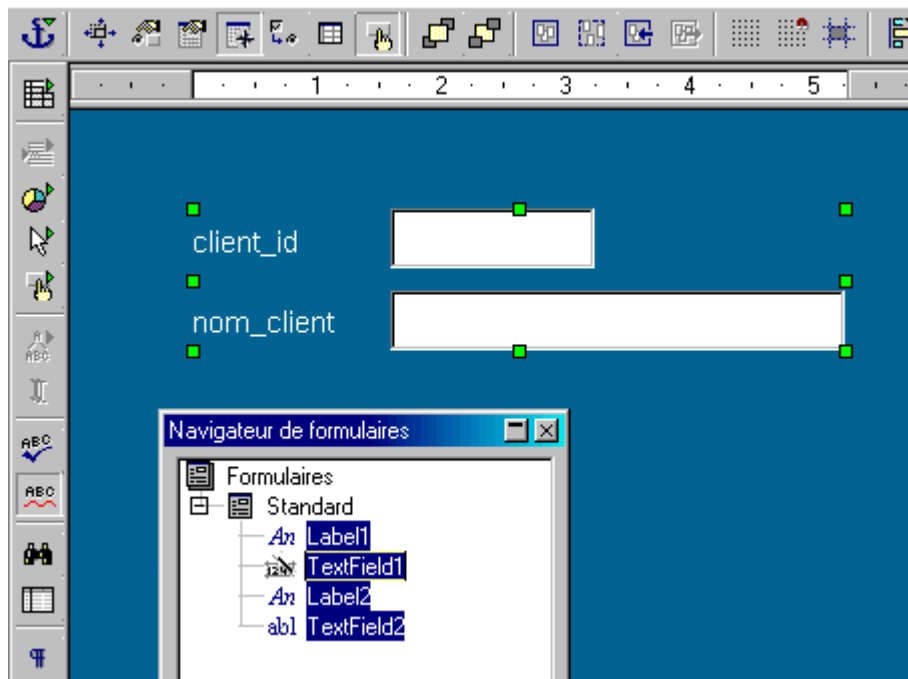
Cliquez sur le bouton **Créer**.

Sélectionnez le répertoire de sauvegarde du formulaire; acceptez le nom suggéré ou introduisez un autre nom. Sauvez enfin le formulaire.

Vous êtes maintenant à la moitié du chemin.

Cliquez sur le bouton du **Navigateur de formulaires** (cinquième bouton à partir de la gauche sur la barre d'outils¹ au dessus du formulaire) . Le navigateur de formulaires montre que vous disposez d'un formulaire, appelé standard, contenant deux labels et deux champs texte (TextFields) .

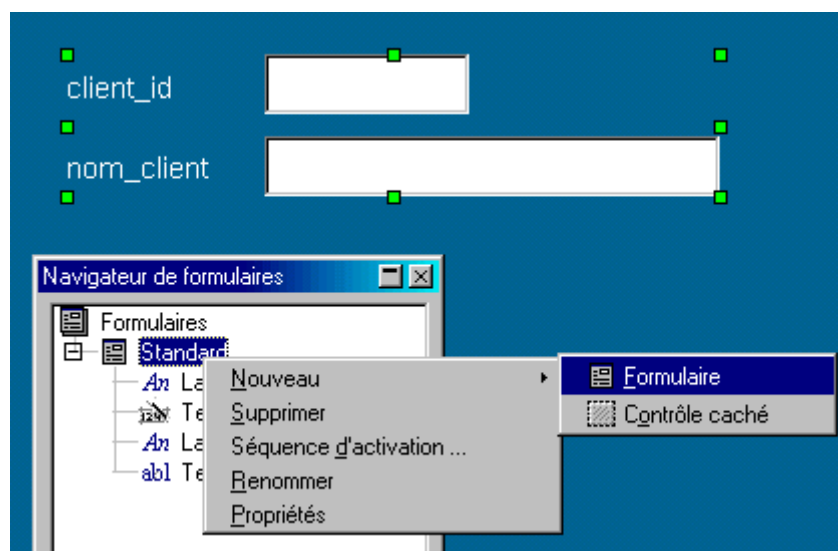
Dans l'illustration ci-dessous, le formulaire Standard est entouré de poignées vertes. Le reste de la page est un simple document Writer (.sxd) .



¹ Attention veillez à être en mode conception !

Ajouter un sous-formulaire

Sélectionnez le formulaire Standard dans le navigateur de formulaires, ensuite cliquez-droit et sélectionnez Nouveau > Formulaire dans le menu contextuel:



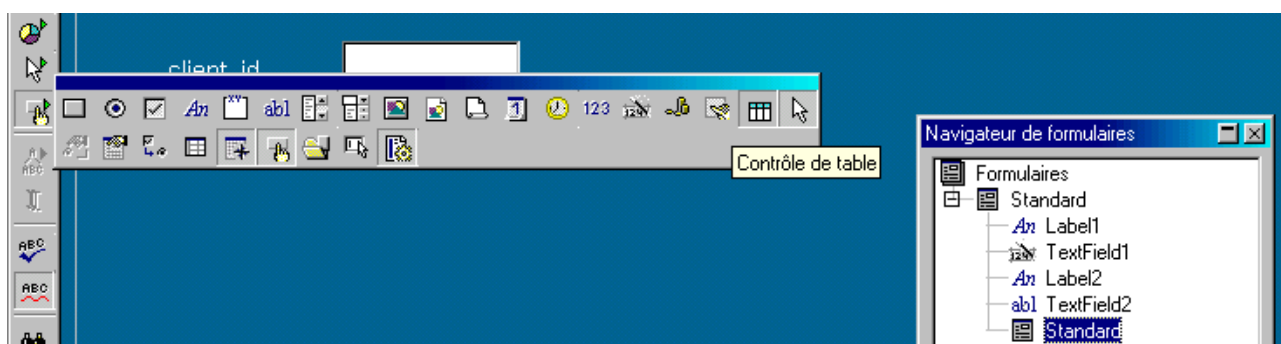
Vous avez maintenant un formulaire maître ou principal et un sous-formulaire. Nous allons lier le formulaire au formulaire principal en quelques minutes au moyen d'une requête SQL avec un paramètre nommé.

Vous pouvez en savoir plus sur les sous-formulaires dans l'aide **OOo Aide > Sommaire > Onglet Index** et en tapant comme terme recherché **sous-formulaire**. Consultez les articles **sous-formulaire**; **formulaire** et **sous-formulaire**; **création avec le navigateur de formulaires**.

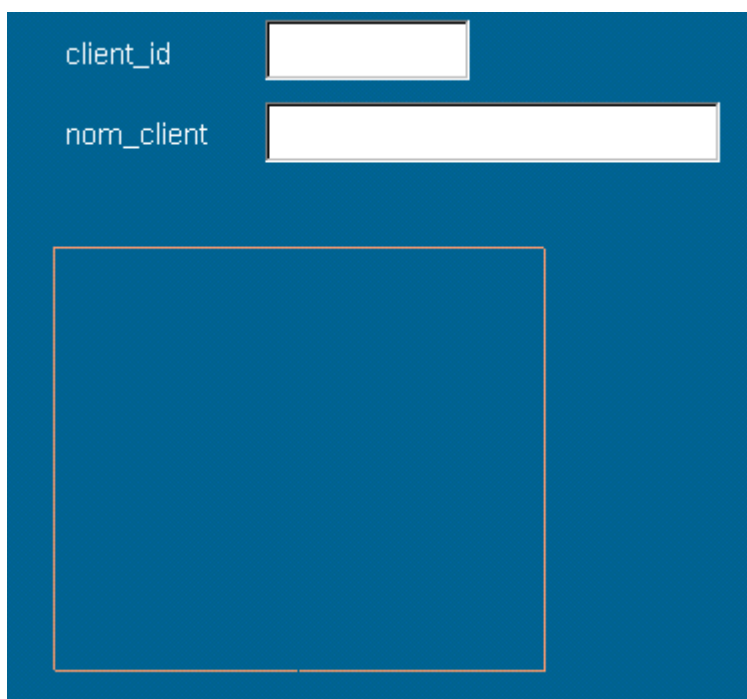
Vous devez maintenant placer un contrôle grille sur la page afin de représenter le sous-formulaire. Assurez vous que le **sous-formulaire Standard** est bien sélectionné dans le navigateur de formulaires, la grille appartiendra, de cette manière, au sous-formulaire et non au formulaire principal. Cliquez-et-maintenez (clique-long) sur le bouton **Formulaire** dans la barre d'outils de gauche (un doigt qui pointe sur un bouton) .

(De manière similaire, un clique-long sur des boutons portant un petit triangle vert, comme Nouveau, Coller, Défaire, Insérer Champs, Insérer Objets, dans la barre d'outils permet d'accéder à une série de possibilités supplémentaires.)

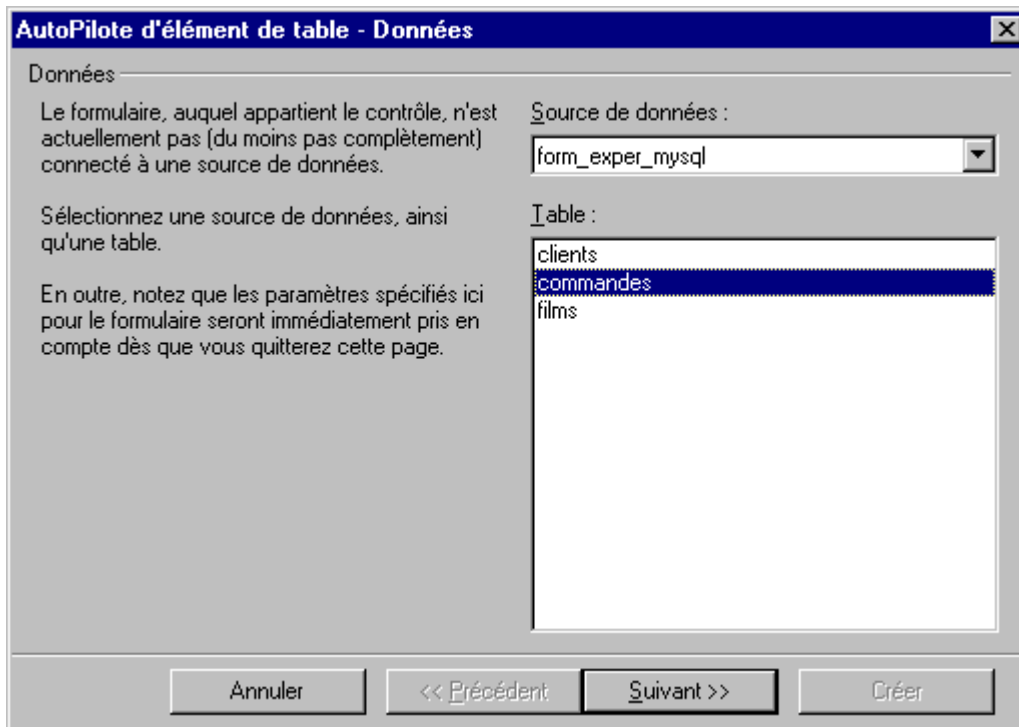
Cliquez sur le bouton **Contrôle de Table**.



Ensuite glissez le curseur afin de déterminer la taille de la grille.

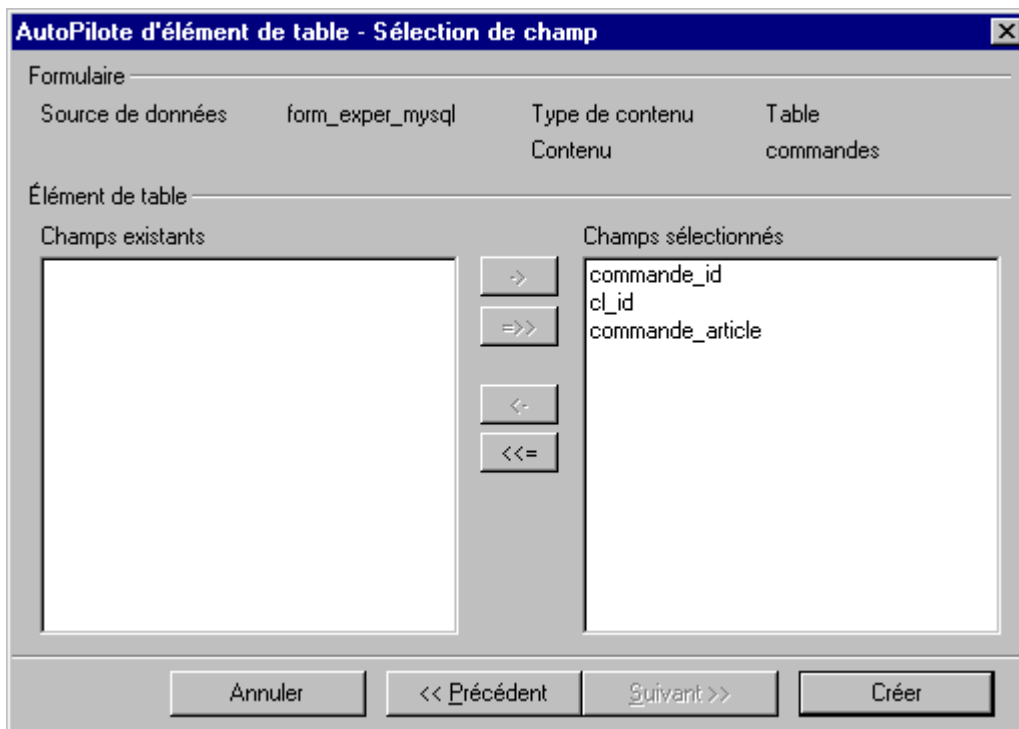


Quand vous relâchez le bouton de la souris, une boîte de dialogue apparaît. Sélectionnez, dans cette boîte de dialogue, la source de données à partir de la liste et sélectionnez ensuite la table **commandes** à partir de la liste des tables disponibles.



Cliquez sur le bouton **Suivant >>**.

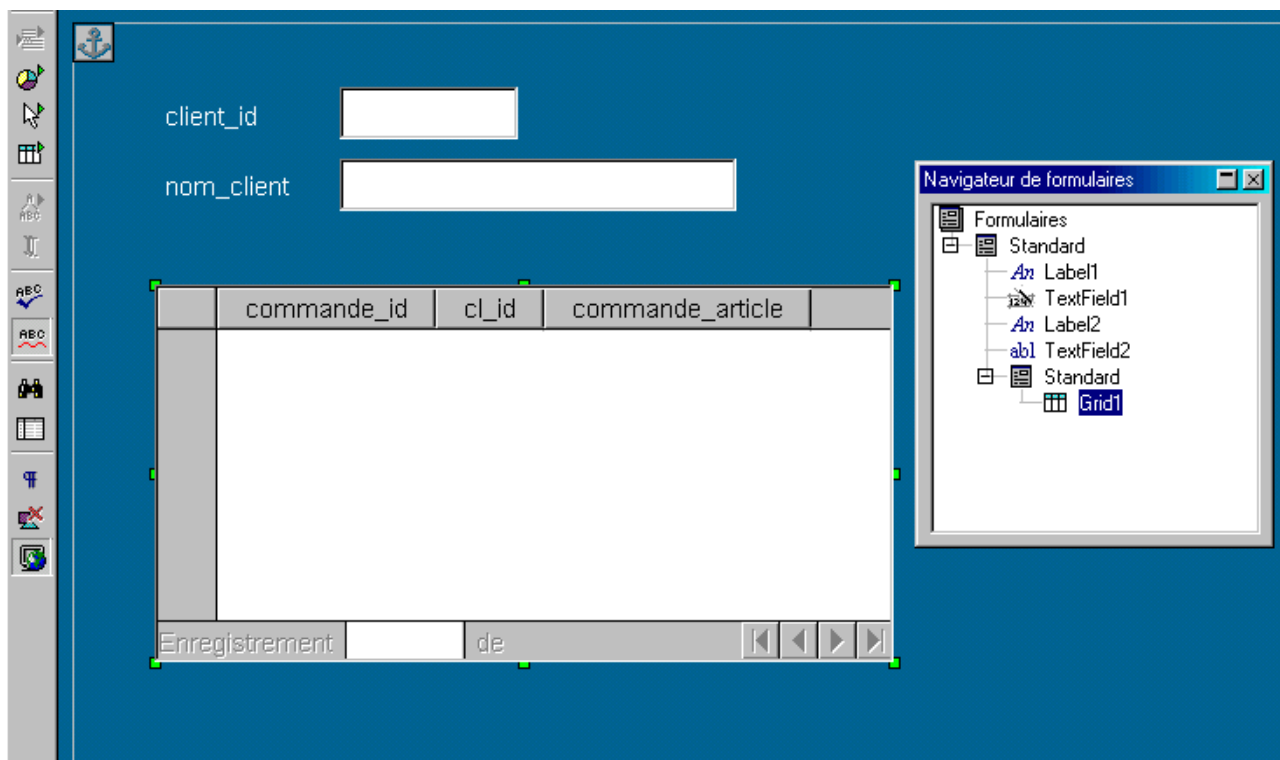
Dans la seconde boîte de dialogue, cliquez sur le bouton **=>>** pour ajouter tous les champs de la table commandes à la grille. Normalement il n'est pas nécessaire d'afficher le champ `cl_id` mais nous le faisons ici à titre d'illustration.



Cliquez sur le bouton **Créer**.

Vous pouvez maintenant adapter la taille de la colonne « commandes_article » pour l'élargir en glissant le coin droit de l'entête de la colonne.

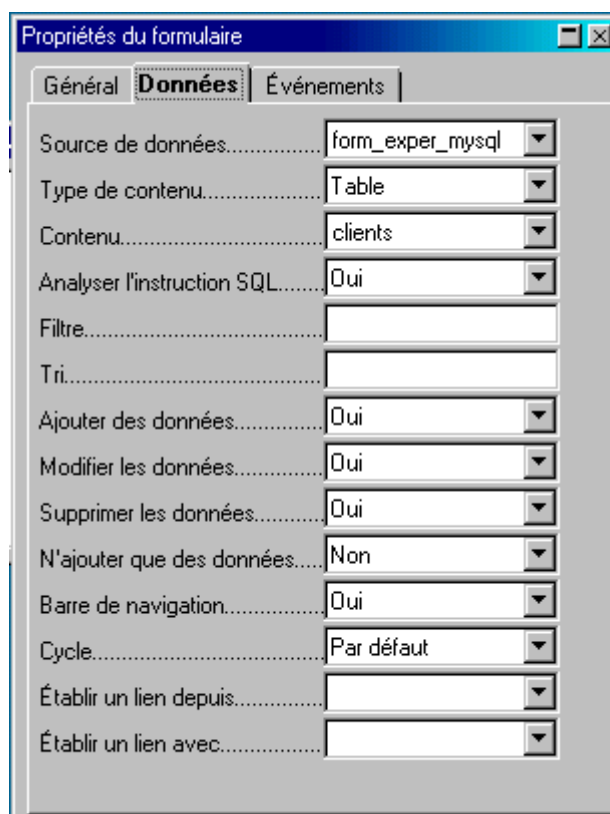
Voici le résultat toujours en mode édition:



Le « mode édition » devrait être appelé plus justement « mode conception de formulaire » afin de distinguer le mode dans lequel le formulaire est créé et le mode dans lequel le formulaire fonctionne, dans lequel vous pouvez éditer les données des tables sous-jacentes. Dans le « mode conception de formulaire », aucune donnée n'est visible mais vous pouvez travailler à la structure et aux propriétés du formulaire.

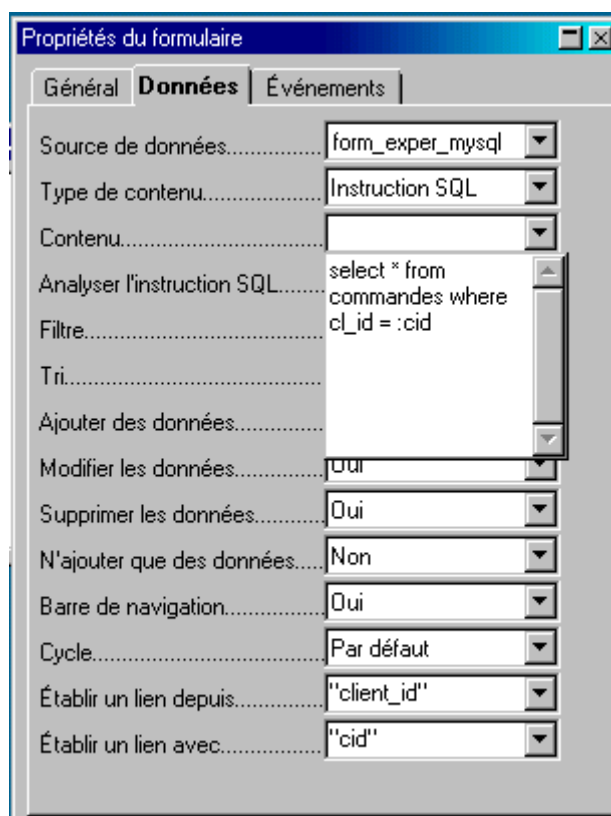
Lier un sous-formulaire à un formulaire principal

Dans le Navigateur de Formulaires, sélectionnez la formulaire principal (le Standard du haut). Cliquez-droit et choisissez Propriétés. L'onglet Données du formulaire devrait ressembler à cela :



Fermez la boîte de dialogue Propriétés.

Dans le Navigateur de Formulaires, sélectionnez maintenant le sous-formulaire (le Standard du dessous). Cliquez-droit et choisissez Propriétés. Adaptez les propriétés dans l'onglet Données afin qu'elles correspondent à l'illustration du dessous :



Assurez-vous que la **Source de données** indique bien votre source de données. Mettez ensuite la valeur **Type de contenu** à « Instruction SQL ».

Cliquez ensuite sur la flèche vers le bas de la propriété **Contenu** et introduisez la requête SQL suivante avec un paramètre nommé :

```
select * from commandes where cl_id = :cid
```

Le paramètre nommé « cid » peut être n'importe quel nom de variable valide précédé par un « : ». Saisissez dans la propriété **Établir un lien depuis** le nom de la clé primaire dans la table clients :

```
client_id
```

N'ajoutez pas les guillemets, le programme le fait pour vous.

Ajoutez ensuite dans la propriété Établir un lien avec le nom de la variable utilisée dans la requête SQL sans le caractère double-point « : » :

`cid`

À nouveau, le programme se charge de mettre les guillemets.

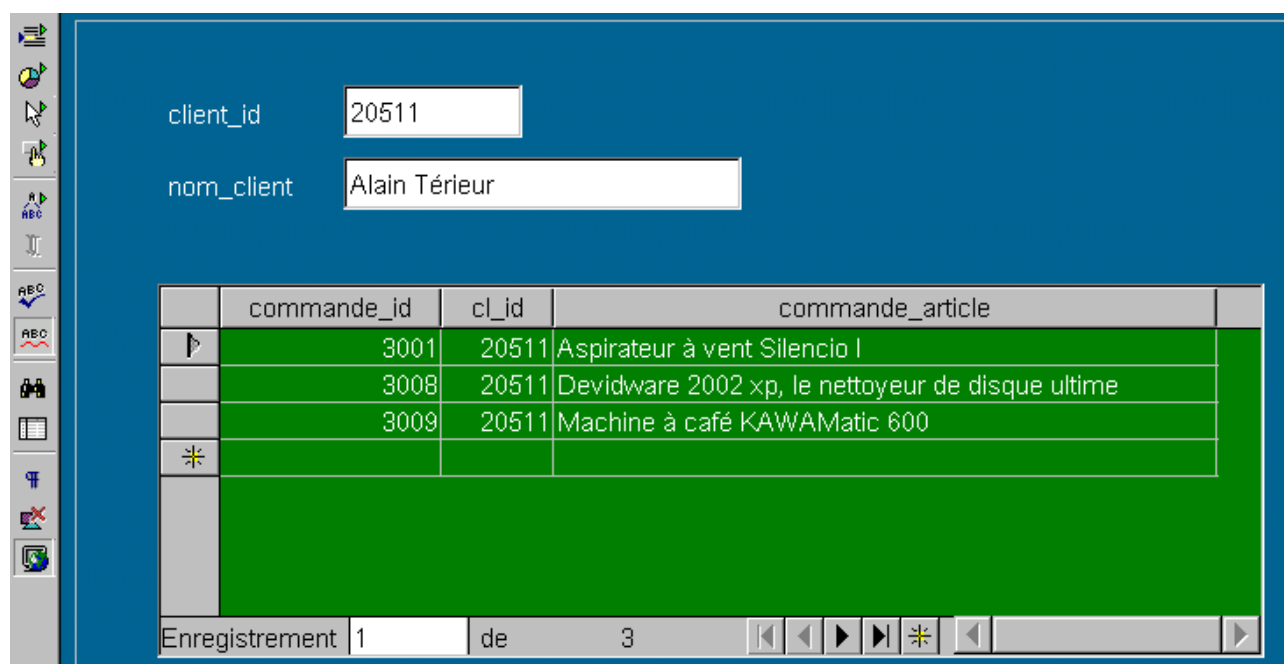
Fermez ensuite la boîte de dialogue.

La variable cid va maintenant prendre la valeur de la clé primaire de l'enregistrement en cours de la table clients. La requête SQL va retourner tous les enregistrements de la table commandes dont le champ cl_id correspond avec la valeur actuelle de cid. Quand vous vous rendez sur le prochain enregistrement de la table clients, cid va prendre une nouvelle valeur.

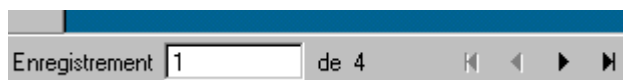
Vous pouvez maintenant changer d'autres propriétés des différents Labels, TextFields et Grids comme les tailles de polices, la couleur de fond, etc.

Quand c'est fait, vous pouvez quitter le mode d'édition en cliquant sur le bouton formulaire . 

Votre formulaire devrait maintenant afficher d'abord le client et ensuite toutes les commandes de ce même client.



Pour se déplacer d'un client à un autre dans la base de données, cliquez sur les boutons du navigateur qui se trouve au bas du formulaire :



Si, au lieu d'observer le résultat escompté, vous recevez un message d'erreur comme le suivant, c'est peut-être à cause du fait que la reconnaissance des paramètres nommés ne s'effectue pas convenablement. Retournez donc en arrière et lancez la macro une nouvelle fois.



Vous devriez être capable d'ajouter de nouvelles commandes pour le client courant ainsi que modifier une commande ou en supprimer une.

client_id	20511
nom_client	Alain Térieur

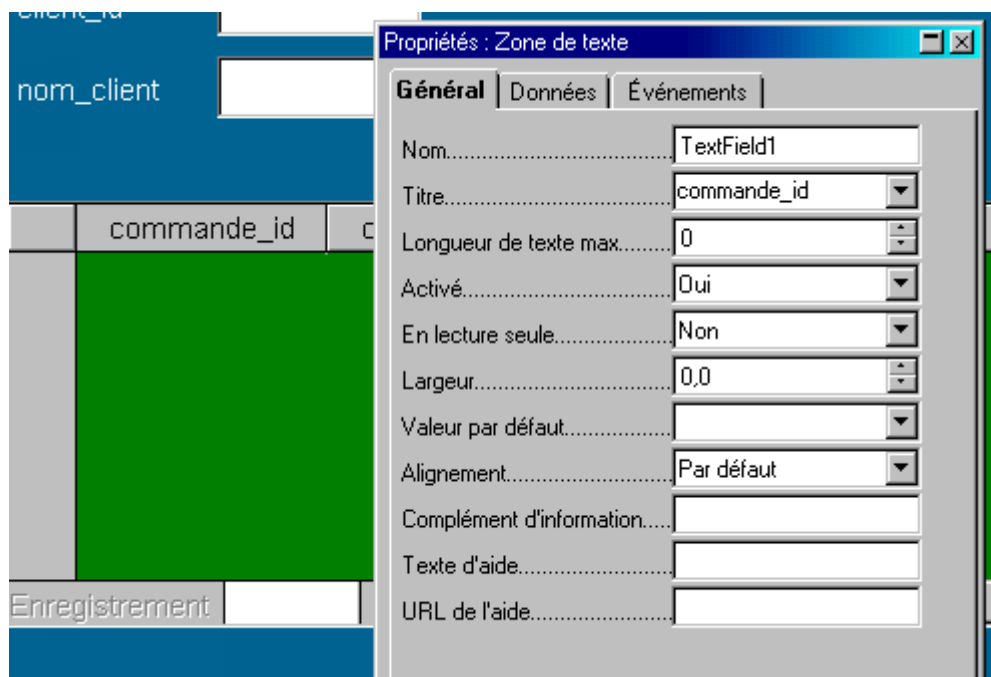
	commande_id	cl_id	commande_article
	3001	20511	Aspirateur à vent Silencio I
	3008	20511	Devidware 2002 xp, le nettoyeur de disque ultime
	3009	20511	Machine à café KAWAMatic 600
	3012	20511	Lave-Linge SA - ESSOR
	3013	20511	C tout pour ce client

Nous avons ajouté ici deux nouvelles commandes pour le client 20511 simplement en les introduisant dans la liste. En fait, dès l'instant où nous n'affichons que les commande pour le client actuel, la valeur du champ cl_id est automatiquement mise à 20511 quand on clique sur la ligne d'insertion d'un nouvel enregistrement, celle marquée par un « * ».

Formatage des champs de données

Si vos champs integer n'apparaissent pas comme des integer, et ressemble par exemple à quelque chose comme 3001.00, c'est parce que OOo crée des champs Zone de texte quand il devrait créer des champs numériques pour les valeurs integer.

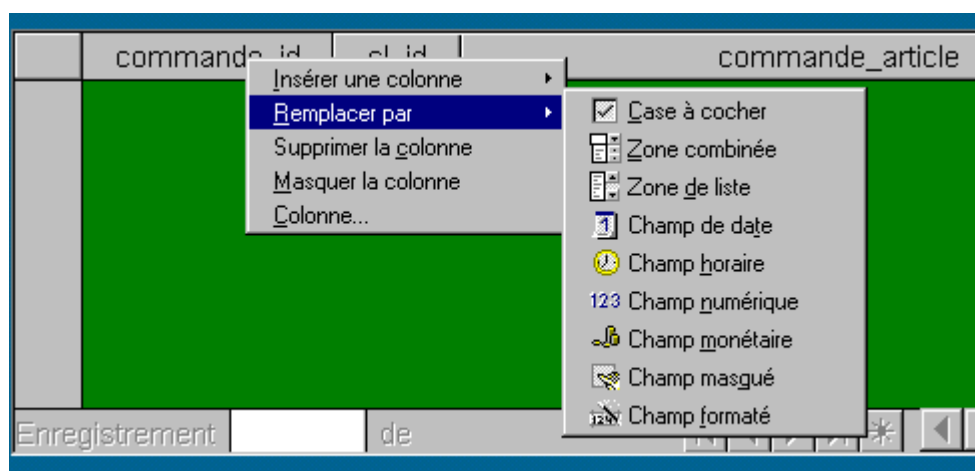
Cliquez sur le bouton Formulaire pour passer en mode édition. Cliquez sur l'entête de la colonne commande_id. Ensuite, cliquez-droit et choisissez **Colonne...** et vous devriez voir la boîte de dialogue suivante:



Notez que vous ne voyez pas apparaître de propriété pour déterminer le nombre de décimales. Nous devons donc avoir un autre type de champ dans notre cas.

(Par héritage, notez également l'existence de la propriété Longueur de texte max.. Quand elle est mise à 0, le nombre maximum de caractères affichés, est déterminé à partir de la taille du champ de la base de données.)

Si vous cliquez-droit une nouvelle fois sur l'entête de la colonne `commande_id`, vous constaterez que vous pouvez insérer des colonnes additionnelles à côté de la colonne `commande_id`, ou vous pouvez aussi remplacer le champ Zone de texte par un champ d'un autre type. Un champ numérique, voire formaté aurait été plus approprié pour l'affichage d'une valeur integer.

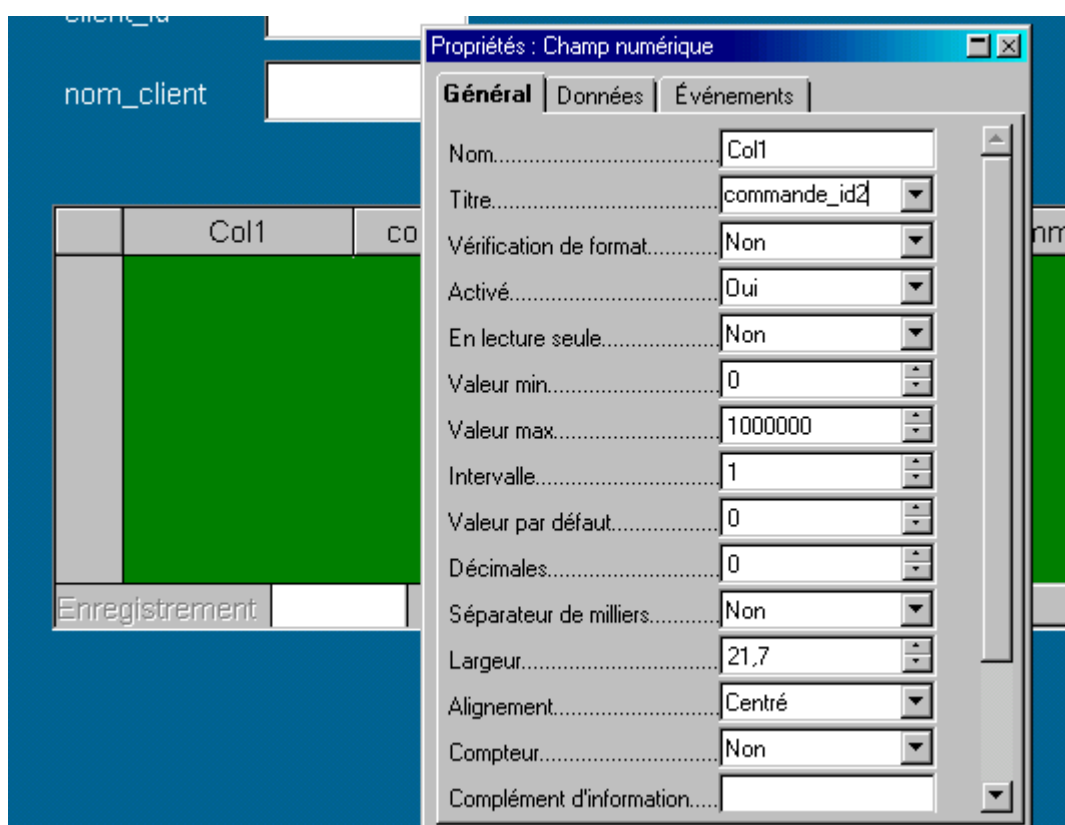


Vous pouvez également déplacer une colonne en utilisant Alt-déplacer.

Expérimentons maintenant ce que nous venons de voir et essayons d'insérer deux colonnes supplémentaires. Comme dans les deux prochaines illustrations, nous allons insérer un champ numérique et un champ formaté.

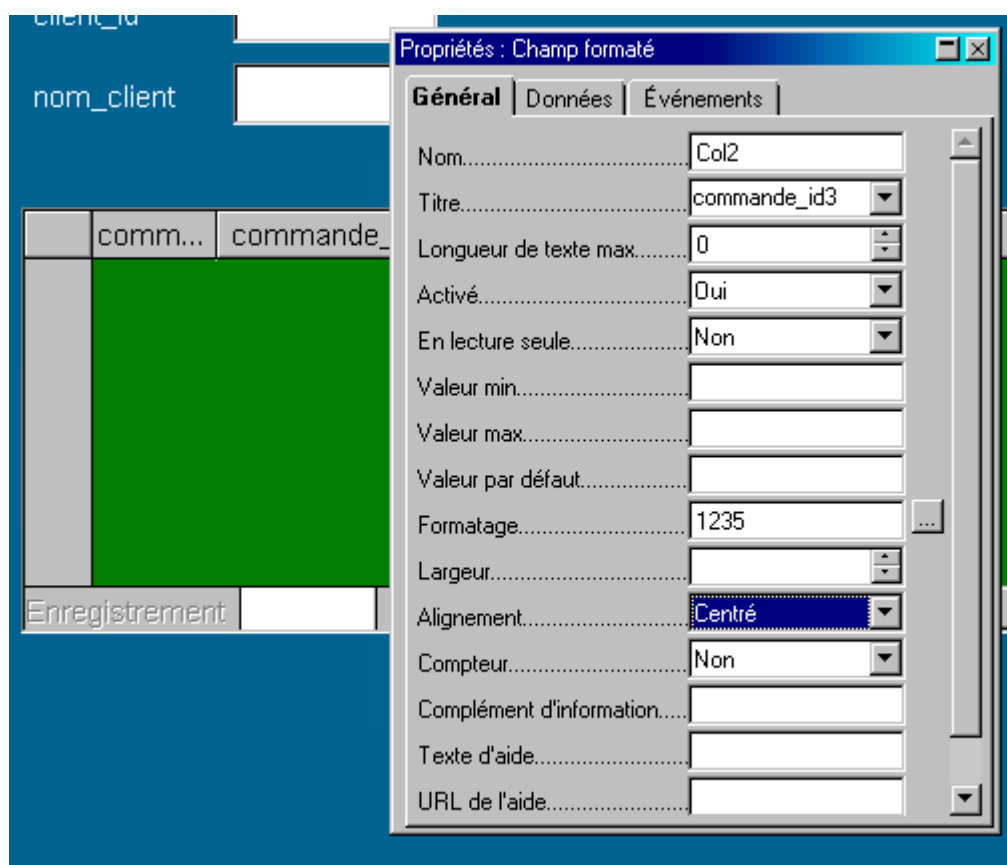
Cliquez sur l'entête de la colonne `commande_id` et choisissez **Insérer Colonne**. Insérez un **champ numérique**, qui sera nommé « Col1 ». Cliquez dessus pour le sélectionner, cliquez-droit ensuite et choisissez **Colonne...** Dans l'onglet Données de la boîte de dialogue, liez le champ au champ `commande_id` de la table `commandes` en introduisant « `commande_id` ».

Dans l'onglet Général, mettez la propriété Titre à « `commande_id2` ». Mettez aussi la propriété Décimales à 0 (aucun chiffre après la virgule). Et comme nous utilisons des valeurs integer non signée, mettez la Valeur min à 0 également. Vous pouvez aussi mettre l'Alignement à « centré ». Fermez ensuite la boîte de dialogue.



Insérez un **champ formaté**, cliquez dessus pour le sélectionner, cliquez-droit ensuite et choisissez **Colonne...** Dans l'onglet Données de la boîte de dialogue, liez le champ au champ commande_id de la table commandes en introduisant « commande_id ».

Dans l'onglet Général, mettez la propriété Titre à « commande_id3 ». Cliquez sur le bouton Formatage et choisissez le format integer dans la liste (celui où aucune décimale n'apparaît). Vous pouvez aussi mettre l'Alignement à « centré ». Fermez ensuite la boîte de dialogue.



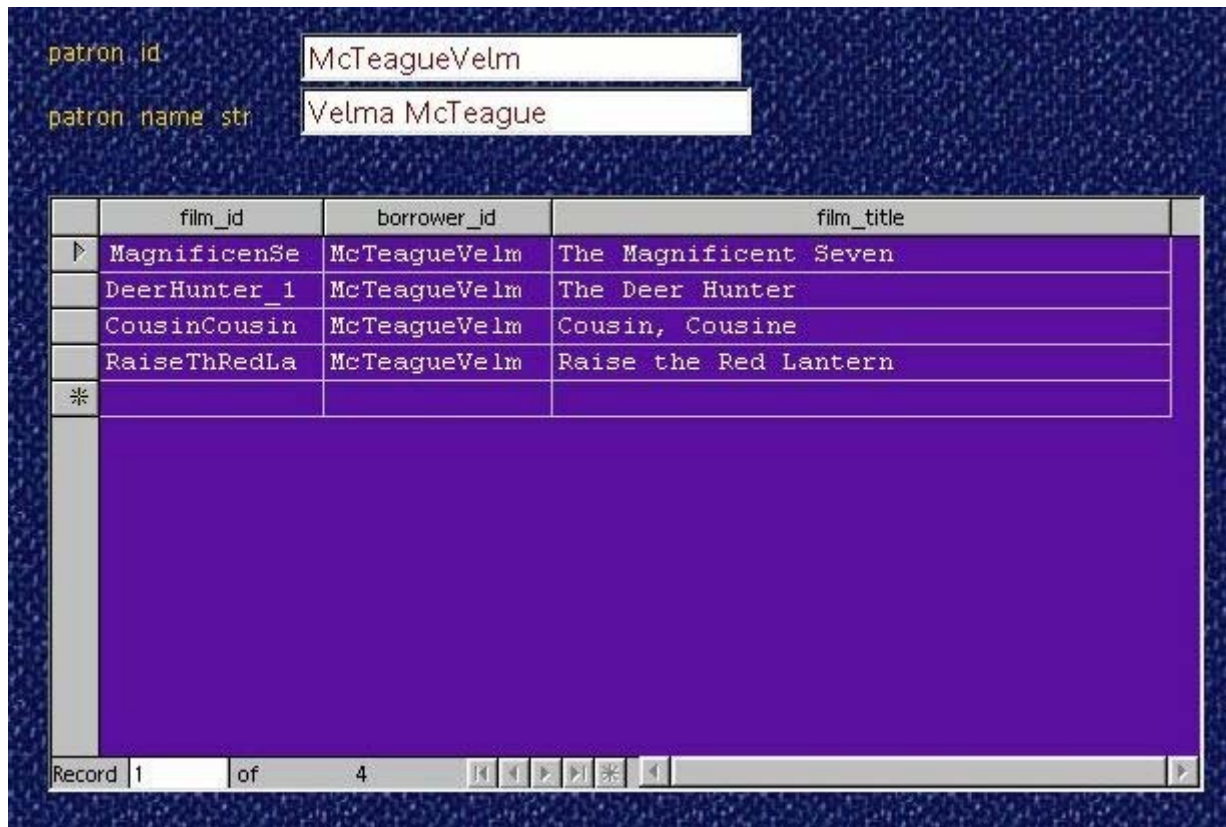
Cliquez sur le bouton Formulaire pour quitter le mode d'édition. Vous devriez maintenant voir trois différents types de champs affichant tous la même valeur commande_id.

Maintenant que vous avez vu le résultat de cette expérience, vous pouvez effacer les deux champs que nous venons juste de créer.

Vous pourriez vouloir remplacer les champs Zone de texte pour les champs commande_id et cli_id par des champs numériques si vos valeurs integer ne s'affichent pas correctement.

Quelques exemples supplémentaires

Cet exemple fonctionne indifféremment avec des clés alphanumériques ou des clés numériques:



The screenshot shows a database application interface. At the top, there are two input fields: 'patron_id' with the value 'McTeagueVelm' and 'patron name str' with the value 'Velma McTeague'. Below these is a table with three columns: 'film_id', 'borrower_id', and 'film_title'. The table contains four rows of data, all with 'McTeagueVelm' in the 'borrower_id' column. A status bar at the bottom indicates 'Record 1 of 4' and includes navigation icons.

	film_id	borrower_id	film_title
▶	MagnificenSe	McTeagueVelm	The Magnificent Seven
	DeerHunter_1	McTeagueVelm	The Deer Hunter
	CousinCousin	McTeagueVelm	Cousin, Cousine
	RaiseThRedLa	McTeagueVelm	Raise the Red Lantern
*			

Record 1 of 4

Ici nous observons deux tables dBase liées de manière analogue:

```
select * from car_color where c_car_id = :cid
```

où `:cid` fait référence à la clé primaire `car_id` dans la table `cars`.

The screenshot shows a dBase table view with a teal background. At the top, there are two input fields: 'CAR_ID' with the value 'ab01' and 'CAR_NAME' with the value 'Toyota Corolla'. Below these is a table with the following data:

	COLOR_ID	C_CAR_ID	COLORNAM
▶	c001	ab01	forest green
	c002	ab01	red
	c004	ab01	pea soup green
	c010	ab01	lime green
*			

At the bottom, there is a record navigation bar showing 'Record 1 of 4' and several navigation buttons (back, forward, search, etc.).

Notez que vous pouvez ajouter un nouvel enregistrement avec le bouton d'ajout d'un nouvel enregistrement * ; il est possible d'éditer les données dans le sous-formulaire. L'enregistrement c010 a été ajouté en l'introduisant directement à partir du sous-formulaire.

Cependant, lier d'autres tables dBase présente moins de réussite. Les tables dBase produites dans OOO n'ont pas fonctionné aussi bien que des tables issues d'autres programmes de base de données et ensuite exportées. La principale différence semble résider dans la faculté de créer la clé primaire.

Voyons maintenant un exemple un petit peu plus complexe avec des tables MySQL. Nous voulons afficher les acteurs qui sont apparus dans un film particulier. Nous avons trois tables : films, actors, et acted_in. La table acted_in contient des paires film_id/actor_id ainsi qu'un nombre séquence appelé billing et un nom de personnage pour chaque rôle dans le film. La plupart des clés sont alphanumériques.

```
create table films (  
    film_id char(12) not null primary key,  
    film_title char(50),  
    rating char(5),  
    genre char(22) );  
  
create table actors (  
    actor_id char(12) not null primary key ,  
    actor_first_name char(16),  
    actor_last_name char(20) );  
  
create table acted_in (  
    acted_in_id int unsigned not null primary key,  
    film_id char(12) not null references films.film_id,  
    actor_id char(12) not null references actors.actor_id,  
    billing smallint unsigned,  
    character_name char(60) );
```

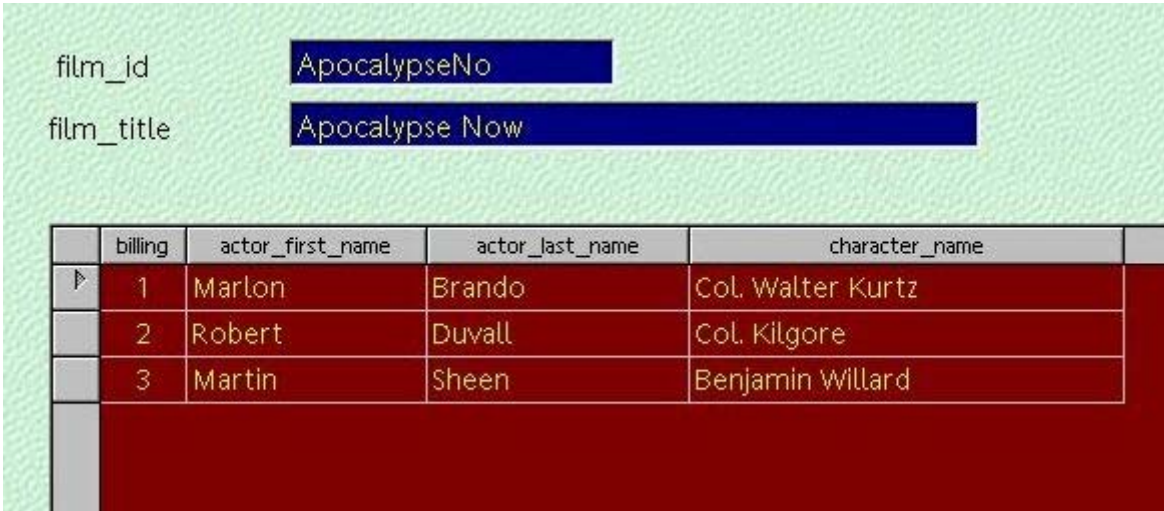
Dans le sous-formulaire, la requête SQL qui lie les tables ensemble est la suivante:

```
select
    acted_in.film_id,
    acted_in.actor_id,
    acted_in.billing,
    actors.actor_first_name,
    actors.actor_last_name,
    acted_in.character_name
from
    acted_in,
    actors
where
    acted_in.film_id = :fid and
    acted_in.actor_id = actors.actor_id;
```

Et le paramètre nommé :fid est lié à la valeur de film_id en cours dans la table films.

Comme ces tables peuvent facilement devenir très larges, vous pourriez avoir besoin d'indexer certaines colonnes afin de rendre l'exécution des requêtes plus rapides. De même, pour la simplicité, nous avons omis d'ajouter à notre exemple une table des directeurs, une tables des scénaristes, et peut-être également une table contenant des photos, voire des clips. Et beaucoup d'acteurs.

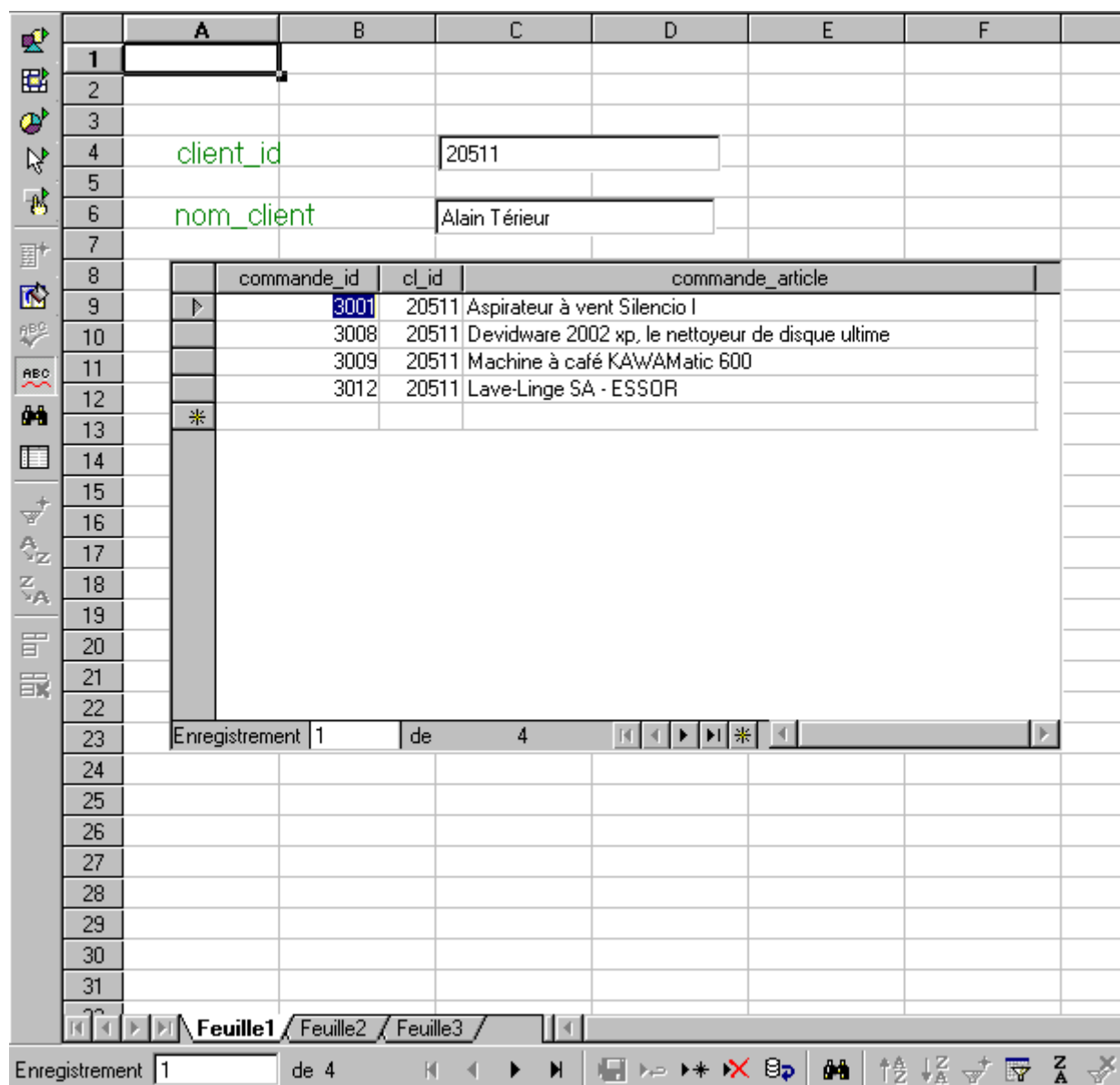
Voici le résultat:



The screenshot shows a database interface with a search bar for 'film_id' containing 'ApocalypseNo' and a corresponding 'film_title' of 'Apocalypse Now'. Below this, a table displays the results of the SQL query, listing actors and their roles in the film.

	billing	actor_first_name	actor_last_name	character_name
▶	1	Marlon	Brando	Col. Walter Kurtz
	2	Robert	Duvall	Col. Kilgore
	3	Martin	Sheen	Benjamin Willard

Les exemples précédents ont été construits à partir de Writer, à l'aide de l'AutoPilot. La même chose est possible à partir d'un classeur, mais vous devrez le faire à la main car aucun AutoPilot n'est disponible.



Crédits

Auteur : Daniel Strome

Remerciements : Aux contributeurs des listes de diffusion users@openoffice.org et dev@dba.openoffice.org.

Intgr Par :

Dernière modification : 24 août 2002 (version original en anglais)
08 janvier 2003 (traduction)

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

Traduction : Raphael Bolle (Texte et illustrations) – Sophie Gauthier (relecture)

Licence

This document is published under the terms and conditions of the Public Documentation License

© 2002 Daniel Strome

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 "How to Link Tables Using SQL Named Parameters".

The Initial Writer of the Original Documentation is Daniel Strome, Copyright © 2002. All Rights Reserved. (Initial Writer contact(s): <mailto:danstrome@openoffice.org>).

Contributor(s): Raphael Bolle (translator - traducteur).

Portions created by Raphael Bolle are Copyright (C) 2002. All Rights Reserved. (Contributor contact(s): raphael_bolle@skynet.be).

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.