LibreOffice

Guide Base

Chapitre 4
Formulaires

Droits d’auteur

Ce document est protégé par Copyright © 2020 par l’Équipe de Documentation de LibreOffice. Les contributeurs sont nommés ci-dessous. Vous pouvez le distribuer et/ou le modifier sous les termes de la Licence Publique Générale GNU (https://www.gnu.org/licenses/gpl.html), version 3 ou ultérieure, ou de la Licence Creative Commons Attribution (https://creativecommons.org/licenses/by/4.0/), version 4.0 ou ultérieure.

Toutes les marques déposées citées dans ce guide appartiennent à leurs légitimes propriétaires.

Contributeurs

De cette édition

Pulkit Krishna
Dan Lewis

Des éditions précédentes

Pulkit Krishna
Robert Großkopf
Hazel Russman
Dan Lewis
Jost Lange
Jean Hollis Weber
Jochen Schiffers

Traduction

Jean-Michel Coste

Relecteurs

Francis Lecher
Patrick AUCLAIR

Retour d’information

Veuillez adresser tout commentaire ou suggestion concernant ce document à la liste de diffusion de l’Équipe de Documentation : doc@fr.libreoffice.org

Note :

Tout ce que vous envoyez à la liste de diffusion, y compris votre adresse mail et toute autre information personnelle incluse dans le message, est archivé publiquement et ne peut pas être effacé.

Date de publication et version du logiciel

Publié en avril 2021. Basé sur LibreOffice 6.4.

Table des matières

Droits d'auteur....2
Contributeurs....2
De cette édition....2
Des éditions précédentes....2
Traduction....2
Relecteurs....2
Retour d'information....2
Date de publication et version du logiciel....2
Les formulaires facilitent la saisie des données....5
Créer des formulaires....5
Un formulaire simple....5
Barres d'outils pour la conception (Ébauche) de formulaires....6
Configurer un formulaire avec le Navigateur de formulaires....6
Créer un formulaire à l'aide d'un champ de formulaire....8
Formulaires externes....9
Propriétés du formulaire....9
Propriétés des contrôles....13
Paramètres par défaut pour de nombreux contrôles....14
Contrôle de texte....20
Contrôle numérique....21
Contrôle de date....21
Contrôle Horaire....22
Contrôle monétaire....22
Contrôle formaté....23
Contrôle Zone de Liste....24
Contrôle Zone combinée....28
Case à cocher....30
Bouton radio....31
Contrôle picto....32
Contrôle de motif....33
Contrôle de table....34
Contrôle Étiquette....35
Zone de groupe....36
Bouton poussoir....38
Bouton picto....39
Barre de navigation....39
Compteurs et barres de défilement....41
Contrôle masqué....42
Sélection multiple....43
Un formulaire simple rempli....44
Ajout de groupes de champs....45
Ajustement des proportions de champ....46
Ajout de champs uniques....49
Contrôle de table....50
Formulaires principaux et sous-formulaires....54
Une vue – plusieurs formulaires....67
Messages d'erreur lors de la saisie dans les formulaires....74

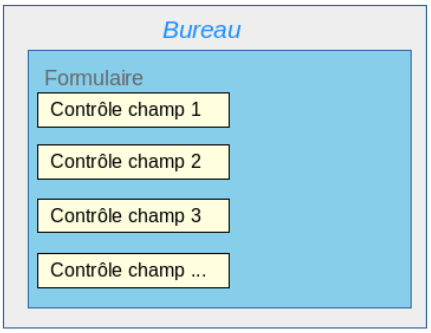
Recherche et filtrage dans les formulaires à l'aide de la barre de navigation.....75
Recherche d'enregistrement à l'aide de paramètres.....75
Filtrer avec l'autofiltre.....76
Filtrage avec le filtre basé sur le formulaire.....77
Filtrer avec le filtre par défaut.....78
Résumé.....80
Saisie d'enregistrements et navigation.....80
Impression à partir de formulaires.....81

Les formulaires facilitent la saisie des données

Les formulaires sont utilisés lorsque la saisie directe dans une table n'est pas pratique, pour détecter rapidement les erreurs de saisie de données, ou lorsque trop de tables rendent la gestion directe des données impossible.

Note

Un formulaire dans Base est une structure invisible pour l'utilisateur. Il sert à permettre l'accès à la base de données. Ce qui est visible pour l'utilisateur est l'ensemble des contrôles, qui servent à la saisie ou à l'affichage de texte, de nombres, etc. Ces contrôles sont divisés par l'interface graphique en différents types.



Le terme *Formulaire* a deux significations. Il peut représenter tout le contenu de la fenêtre de saisie qui est utilisée pour gérer les données d'une ou plusieurs tables. Une telle fenêtre peut contenir un ou plusieurs formulaires principaux, et chacun de ceux-ci peut contenir des sous-formulaires. Le mot formulaire est également utilisé pour ces zones partielles. Il devrait être clair à partir du contexte quel sens est voulu afin d'éviter les malentendus.

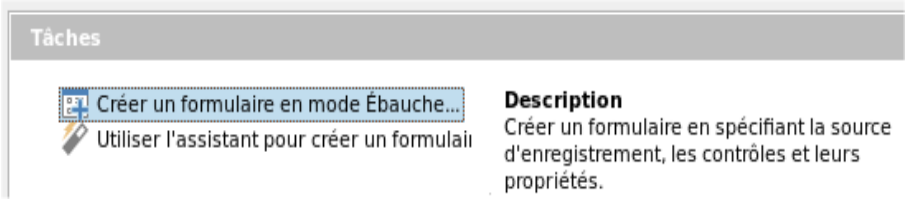
Créer des formulaires

Le moyen le plus simple de créer un formulaire consiste à utiliser l'Assistant Formulaire. Son utilisation pour créer un formulaire est décrite au chapitre 8, *Débuter avec Base*, dans le Guide de mise en route. Ce chapitre explique également comment vous pouvez modifier davantage le formulaire après avoir utilisé l'assistant.

Ce guide présente la création d'un formulaire sans utiliser l'assistant. Il décrit également les propriétés des différents types de contrôles dans un formulaire.

Un formulaire simple

Nous commençons par utiliser la tâche **Créer un formulaire en mode Ébauche** dans la zone Formulaires de la fenêtre principale de base.



Cela appelle l'éditeur de formulaire et le formulaire est affiché dans la fenêtre Vue de création (Figure 1).

La barre d'outils Contrôles de formulaire est ancrée sur le côté gauche. La barre d'outils Ébauche de formulaire (Figure 2) est ancrée en bas. Si ces barres d'outils n'apparaissent pas automatiquement, utilisez **Affichage >Barres d'outils** pour les afficher. Sans ces barres d'outils, il n'est pas possible de créer un formulaire.

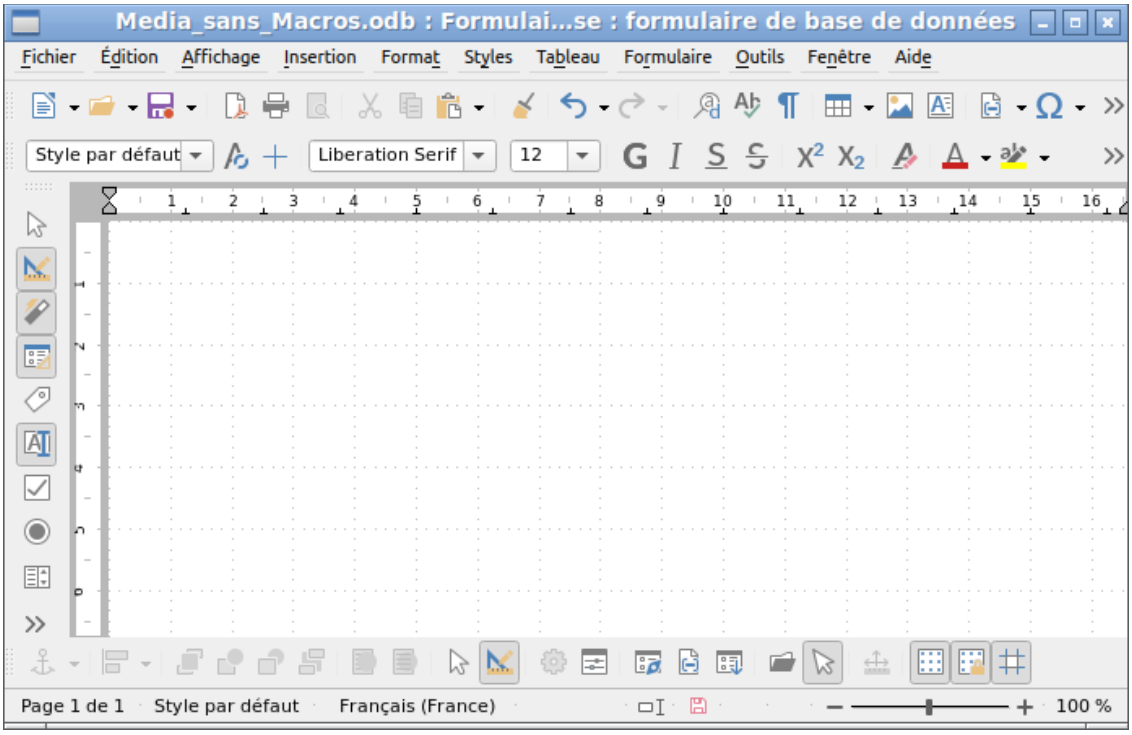


Figure 1: Formulaire affiché en mode Création

La zone vide montre une grille de points. Cette grille vous aide à positionner les contrôles avec précision, en particulier les uns par rapport aux autres. Les symboles à l'extrémité droite de la barre d'outils Création de formulaire indiquent que la grille est visible et active. Les trois derniers symboles (Afficher la grille, Aligner…), doivent être visibles et actifs. Si tous ne le sont pas, cliquez sur ceux qui ne le sont pas.

Barres d'outils pour la conception (Ébauche) de formulaires

Un formulaire est créé sur la page vide. Ceci peut être fait de deux façons :

- Utilisez le Navigateur de formulaires pour configurer un formulaire, ou
- Concevez les contrôles de formulaire et configurez le formulaire à l'aide du menu contextuel.

Configurer un formulaire avec le Navigateur de formulaires

Pour afficher le navigateur de formulaires, cliquez sur le bouton Navigateur de formulaires (illustré à la Figure 2). Une fenêtre apparaît (Figure 3), elle, n'affiche qu'un seul dossier, intitulé Formulaires. Il s'agit du niveau le plus élevé de la zone que nous éditons. Plusieurs formulaires peuvent être accueillis ici.

Figure 2: Barre d'outils Ébauche (conception) de formulaires.

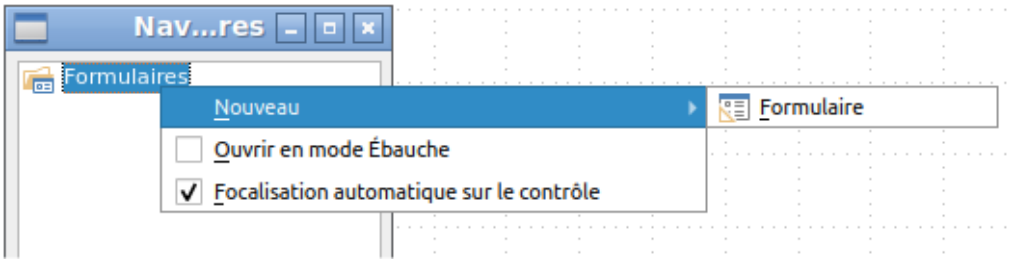


Figure 3: Utilisation du navigateur de formulaires pour créer un nouveau formulaire

Dans le navigateur de formulaires (Figure 3), cliquez avec le bouton droit sur Formulaires pour ouvrir un menu contextuel. Choisissez **Nouveau >Formulaire** pour créer un nouveau formulaire. Les autres choix du menu contextuel (Ouvrir en mode Ébauche et Focalisation automatique sur le contrôle) correspondent aux boutons de la Figure 4, nous en discuterons plus tard.

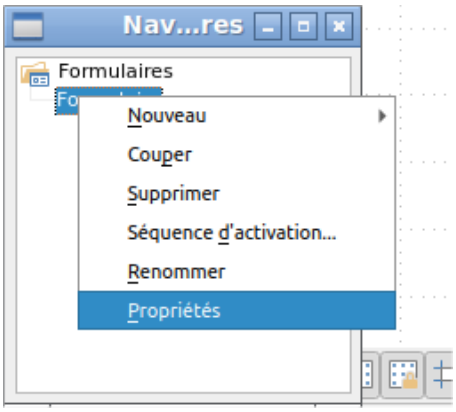
Note

Pour lancer un formulaire avec le curseur dans le premier champ, utilisez l'option Focalisation automatique sur le contrôle. Le premier élément est déterminé par la séquence d'activation du formulaire.

Malheureusement, il y a actuellement un bogue (bogue 87290) dans cette fonction. Si un formulaire contient un contrôle de table, le focus est défini automatiquement sur le premier champ de ce contrôle. Curieusement, ce bogue est corrigé si, après avoir choisi le Focalisation automatique sur le contrôle, vous changez la langue de l'interface utilisateur.

Le formulaire porte le nom par défaut Formulaire. Vous pouvez modifier ce nom immédiatement ou ultérieurement. Cela n'a aucune signification sauf si vous devez accéder à une partie du formulaire à l'aide de macros. La seule chose dont vous devez vous assurer est que deux éléments portant le même nom n'apparaissent pas au même niveau dans l'arborescence des dossiers.

Le menu contextuel du formulaire (illustré ci-dessous) permet de créer des propriétés de formulaire.



Créer un formulaire à l'aide d'un champ de formulaire

La barre d'outils Contrôles de formulaire (Figure 4) contient les boutons avec lesquels les contrôles (étiquette plus champ) peuvent être créés. Tous les boutons ne sont pas directement visibles sur le côté gauche du formulaire. Cliquez sur le bouton du bas (**»**) pour voir le reste des boutons visibles. Pour voir tous les boutons, visibles ou non, cliquez avec le bouton droit de la souris sur n'importe quelle partie de la barre d'outils et sélectionnez **Boutons visibles**. Les boutons visibles sont cochés (icône grisée), les non affichés ne le sont pas (icône sur fond blanc) (Figure 5).

Cela permet de sélectionner les boutons que l'utilisateur souhaite utiliser régulièrement tout en supprimant les indésirables. Par exemple, si le bouton Contrôle de table n'est pas visible, mais peut être activé en le localisant d'abord dans la liste de tous les boutons, puis en le cochant.

Figure 4: Contrôles de formulaire

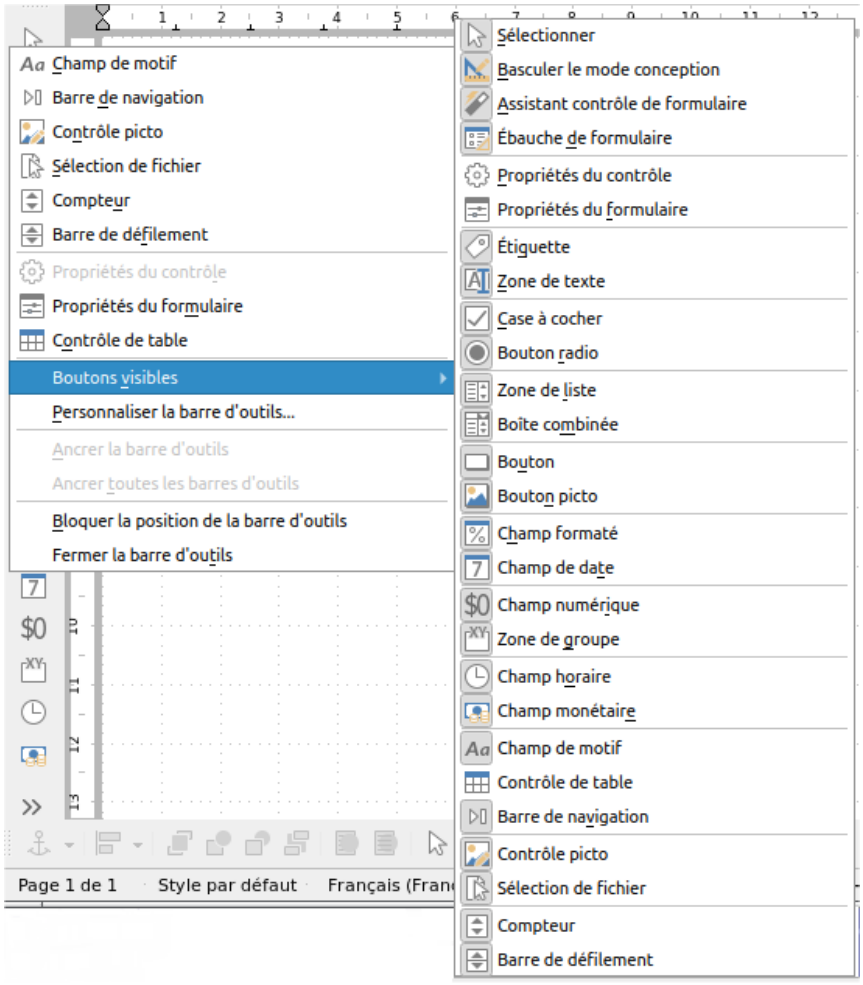


Figure 5: Choix des boutons visibles

Lorsque vous sélectionnez un contrôle de formulaire, vous créez automatiquement un formulaire. Par exemple, supposons que vous choisissiez un champ de texte : le curseur change de forme et une forme rectangulaire peut être dessinée sur la surface blanche du formulaire. Ensuite, sur la surface pointillée du formulaire, un champ de texte apparaît.

Vous pouvez maintenant créer le formulaire en cliquant avec le bouton droit de la souris et en utilisant le menu contextuel du contrôle (Figure 6). Sur cette figure, on a choisi de dessiner un champ d'étiquette.

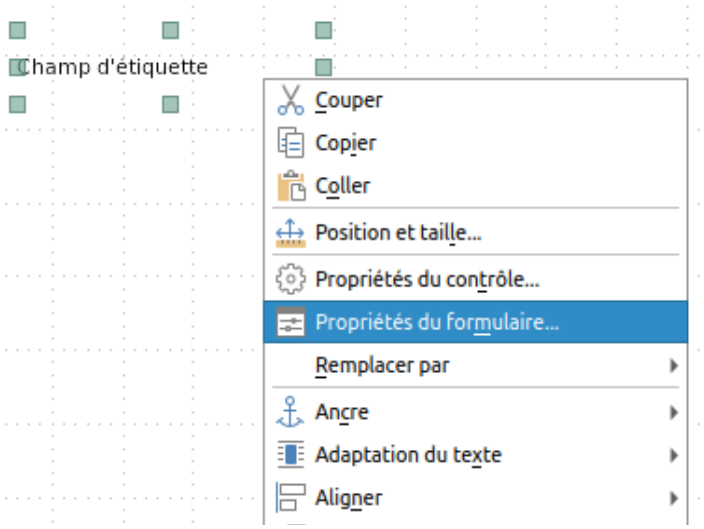


Figure 6: Menu contextuel de contrôle

Sélectionnez l'option **Propriétés de formulaire** (mise en évidence dans l'illustration) pour définir les propriétés du formulaire que vous venez de créer. Le formulaire a le nom par défaut Formulaire.

Il est également possible d'accéder aux propriétés du formulaire avec le bouton prévu dans la barre d'outils de conception (s'il est visible).

Formulaires externes

Outre les formulaires créés dans Base, il est également possible de créer des formulaires dans Writer ou Calc. Ceux-ci sont décrits au Chapitre 7, Connexions aux bases de données.

Propriétés du formulaire

Lorsque les propriétés du formulaire sont appelées à l'aide du menu contextuel du navigateur de formulaires ou du menu contextuel d'un contrôle de formulaire ou encore de l'icône de la barre d'outils, une fenêtre Propriétés du formulaire apparaît. Il comporte trois onglets : Général, Données et Événements.

Onglet Général

Ici, vous pouvez modifier le nom du formulaire. En outre, il existe des possibilités de conception qui n'ont aucune importance à l'intérieur de Base. Elles ne montrent que les possibilités plus générales de conception à l'aide d'un éditeur de formulaires : lorsque vous créez un formulaire Web, vous devrez les utiliser.

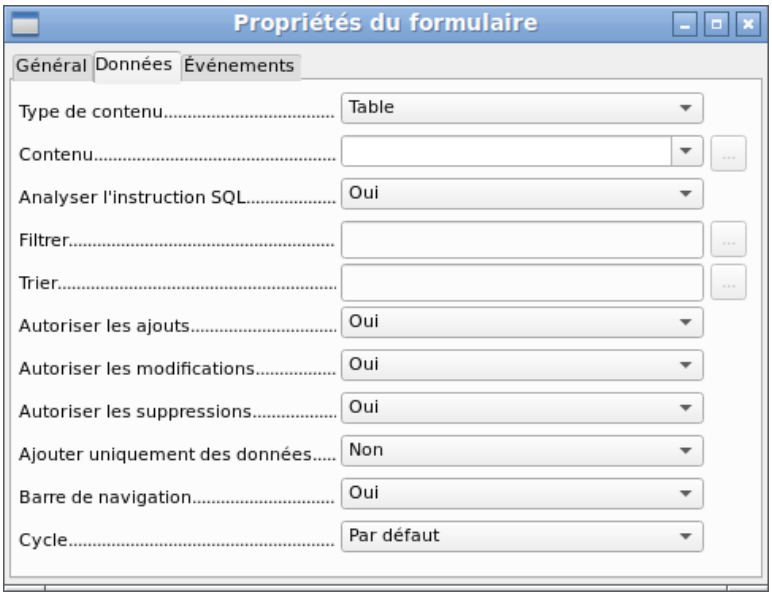
URL : Destination pour les données.

Cadre : Section du site Web de destination à adresser si nécessaire.

Encodage de l'envoi : en plus du codage de caractères normal pour la transmission à l'URL, vous pouvez spécifier ici le codage de texte et le codage en Multipart (par exemple, pour le transfert de données).

Type d'envoi : GET (*Obtenir* visible via l'URL attachée au nom de fichier, vous pouvez le voir souvent sur le Web si vous utilisez un moteur de recherche) ou POST (non visible, convient aux gros volumes de données). GET et POST sont des méthodes de transmission de données pour les formulaires Web.

Onglet Données



Pour créer des formulaires internes dans Base, il s'agit de l'onglet le plus important. Ici, vous pouvez définir les propriétés initiales suivantes pour le formulaire :

Type de contenu : Choisissez entre les contenus Table, Requête et SQL. Bien que Table puisse toujours être utilisé pour la saisie de données dans un formulaire, ce n'est pas toujours le cas pour Requête (pour plus d'informations, voir le chapitre 5, Requêtes) ou la saisie directe d'une commande SQL. Il s'agit ici d'une requête qui n'est pas visible dans le conteneur de requêtes de Base mais qui a en principe la même structure.

Contenu : Selon que Table ou Requête a été choisie ci-dessus, toutes les tables et requêtes disponibles sont répertoriées. Si une commande SQL doit être créée, vous pouvez appeler l'éditeur de requête en utilisant les points de suspension (…) à droite du champ Contenu.

Analyser la commande SQL : Si l'analyse des commandes SQL ne doit pas être autorisée (parce que, par exemple, vous utilisez du code que l'interface graphique ne peut pas afficher

correctement), vous devez choisir **Non** ici. Cependant, cela empêchera le formulaire d'accéder aux données sous-jacentes à l'aide d'un filtre ou d'un tri.

Filtrer : Ici, vous pouvez définir un filtre. Pour obtenir de l'aide, cliquez sur le bouton à droite du champ. Voir également le Chapitre 3, Tables.

Tri : Ici, vous pouvez configurer un tri pour vos données. Pour obtenir de l'aide, cliquez sur le bouton à droite du champ. Voir également le Chapitre 3, Tables.

Autoriser les ajouts : L'entrée de nouvelles données est-elle autorisée ? Par défaut, il est défini sur **Oui**.

Autoriser les modifications : L'édition des données est-elle autorisée ? Par défaut également **Oui**.

Autoriser les suppressions : La suppression des données est également autorisée par défaut.

Ajouter uniquement des données : Si vous choisissez **OUI** pour cette option et **Non** pour les autres sélections, un formulaire vide sera toujours affiché. Il n'y aura aucun accès aux données existantes, qui ne peuvent être ni modifiées ni visualisées. Cependant, si **Autoriser les ajouts** et **Ajouter uniquement des données** est sélectionné, les données peuvent être ajoutées à un champ. Mais une fois que les données ont été enregistrées, elles ne sont plus visibles et les données sont écrites dans les tables.

Conseil

Cette propriété peut être utile lorsque des données doivent être ajoutées à la base de données mais que la personne qui le fait n'est pas autorisée à modifier ou supprimer des données existantes. Ces actions étant laissées à la charge d'une autre personne qui utilise pour cela un formulaire différent offrant ces possibilités.

Barre de navigation : La visibilité de la barre de navigation en bas de l'écran peut être activée ou désactivée. Il est également possible, lorsque vous avez un sous-formulaire, de toujours afficher la barre de navigation pour le formulaire principal, de sorte que l'activation de cette barre d'outils n'affecte que le formulaire principal. Ce paramètre de la barre de navigation n'est pas pertinent pour la barre d'outils de navigation interne qui peut être ajoutée en tant que contrôle de formulaire si nécessaire. Celle ci ne servira que dans le formulaire (ou sous formulaire) dans lequel elle se trouve.

Cycle : L'option par défaut pour les bases de données Base est qu'après l'entrée dans le dernier champ d'un formulaire, la touche Tabulation vous amène au premier champ de l'enregistrement suivant – c'est-à-dire qu'un nouvel enregistrement sera créé. Pour les bases de données, cela a le même effet que ***Tous les enregistrements*.** En revanche, si vous choisissez **Enregistrement actif**, le curseur se déplacera uniquement dans l'enregistrement ; lorsqu'il atteint le dernier champ, il reviendra au premier champ de cet enregistrement. **La page actuelle** fait référence en particulier aux formulaires HTML. Le curseur saute de la fin d'un formulaire au formulaire suivant sur cette page plus bas

Onglet Événements

Événement peut déclencher des macros. Un clic sur le bouton à droite (…) permet d'associer des macros à l'événement

Réinitialisation/Rétablissement : Le formulaire est vidé de toutes les nouvelles entrées qui n'ont pas encore été enregistrées.

Avant l'envoi : Avant l'envoi des données du formulaire. Cela n'a de sens que pour les formulaires Web.

Avant le rechargement : Uniquement lors de l'ouverture du formulaire, pas lors du chargement d'un nouvel enregistrement dans le formulaire.

Rechargement : Cela se produit lorsque le contenu du formulaire est actualisé, par exemple à l'aide d'un bouton dans la barre de navigation.

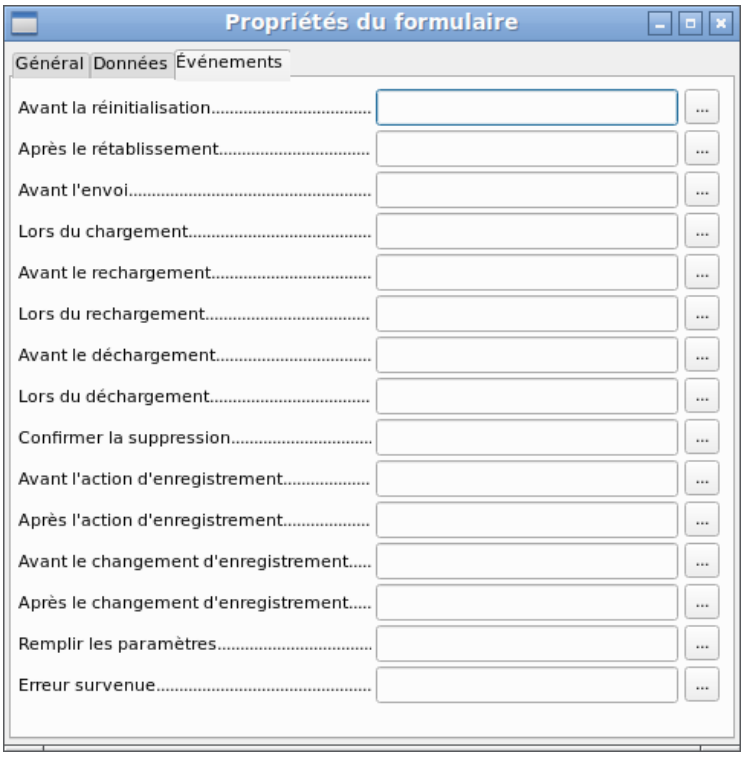
Déchargement : Cette option ne semble pas fonctionner. Il devrait se référer à la fermeture du formulaire.

Confirmer la suppression : L'événement **Confirmation de suppression** se produit dès que les données ont été supprimées du formulaire. Par exemple, la macro liée peut demander une confirmation dans une boîte de dialogue.

Action d'enregistrement : Cela inclut, par exemple, le stockage à l'aide d'un bouton. Dans les tests, cette action se duplique régulièrement ; les macros s'exécutent deux fois de suite.

En effet, ici différentes fonctions ("implémentations") sont effectuées. Les deux sont nommées : `org.openoffice.comp.svx.FormController` et `com.sun.star.comp.forms.ODatabaseForm`. Si dans la macro qui utilise `oForm.ImplementationName`, le nom correspondant est interrogé, la macro peut être limitée à une exécution.

Changement d'enregistrement : l'ouverture d'un formulaire compte comme un changement d'enregistrement. Chaque fois qu'on passe à d'un enregistrement à un autre dans le formulaire, cette action se produit également deux fois. Les macros sont donc exécutées deux fois de suite. Ici aussi, nous pouvons distinguer les causes de ce résultat.



Remplir les paramètres : Cette macro sera exécutée si une requête de paramètre doit être appelée dans un sous-formulaire, mais pour une raison quelconque, le paramètre n'est pas correctement transmis à partir du formulaire principal. Si cet événement n'est pas intercepté, une requête de paramètre suivra le chargement du formulaire.

Erreur survenue : L'événement **Erreur survenue** est activé si une erreur se produit lors de l'accès à la source de données. Cela s'applique aux formulaires, aux zones de liste et aux boîtes combinées

Propriétés des contrôles

Une fois qu'un formulaire a été créé, il peut être rempli de contrôles visibles. Certains contrôles permettent d'afficher le contenu de la base de données ou de saisir des données dans la base. D'autres contrôles sont utilisés exclusivement pour la navigation, pour la recherche et pour l'exécution de commandes (interaction). Certains contrôles servent à une retouche graphique supplémentaire du formulaire.

Saisie et affichage des données	
Contrôle	***Usage***
Zone de texte	Saisie de texte
Champ numérique	Saisie de nombres
Champ de date	Saisie de dates
Champ horaire	Saisie des heures
Champ monétaire	Saisie numérique, préformée pour la devise locale
Champ formaté	Affichage et saisie avec mise en forme supplémentaire, par exemple à l'aide d'unités de mesure
Zone de liste	Choix entre plusieurs possibilités différentes, également pour le transfert dans la base de données de valeurs autres que celles affichées
Zone combinée	Similaire à un champ de liste, mais avec seulement la valeur affichée transférée, où vous pouvez entrer de nouvelles valeurs à la main
Case à cocher	Champ Oui/Non
Bouton radio	Permet de choisir parmi un petit nombre de possibilités
Contrôle picto	Affichage d'images d'une base de données et entrée d'images dans une base de données via une sélection de chemin
Champ de motif	Entrée dans un masque prédéfini ; limite les possibilités de saisie à des combinaisons de caractères spécifiques
Contrôle de table	Module d'entrée universel, qui peut afficher une table entière. Plusieurs des commandes ci-dessus sont intégrées à ce contrôle
Conception	
Contrôle	***Usage***
Étiquette	En-tête du formulaire, description des autres contrôles
Zone de groupe	Un cadre autour, par exemple, d'un ensemble de boutons d'option
Interaction	
Contrôle	***Usage***
Bouton	Bouton avec étiquette
Bouton picto	Comme un bouton, mais avec une image supplémentaire (graphique) affichée dessus
Barre de navigation	Barre d'outils très similaire à celle en bas de l'écran
Sélection de fichier	Pour sélectionner des fichiers, par exemple pour les télécharger dans un formulaire HTML – pas décrit plus en détail
Compteur	Ne peut être utilisé qu'avec une macro – pas de description plus détaillée

Barre de défilement	Ne peut être utilisé qu'avec une macro – pas de description plus détaillée
Contrôle caché	Ici, une valeur peut être stockée à l'aide de macros puis lue à nouveau,

Paramètres par défaut pour de nombreux contrôles

Comme pour les formulaires, les propriétés de contrôle (Cf. Fig 7) sont regroupées en trois catégories : Général, Données et Événements.

L'onglet Général comprend tout ce qui est visible pour l'utilisateur.

L'onglet Données spécifie la liaison à un champ de la base de données ou une requête SQL.

L'onglet Evenements contrôle les actions, qui peuvent être liées à une macro. Dans une base de données sans macros, cette catégorie ne joue aucun rôle.

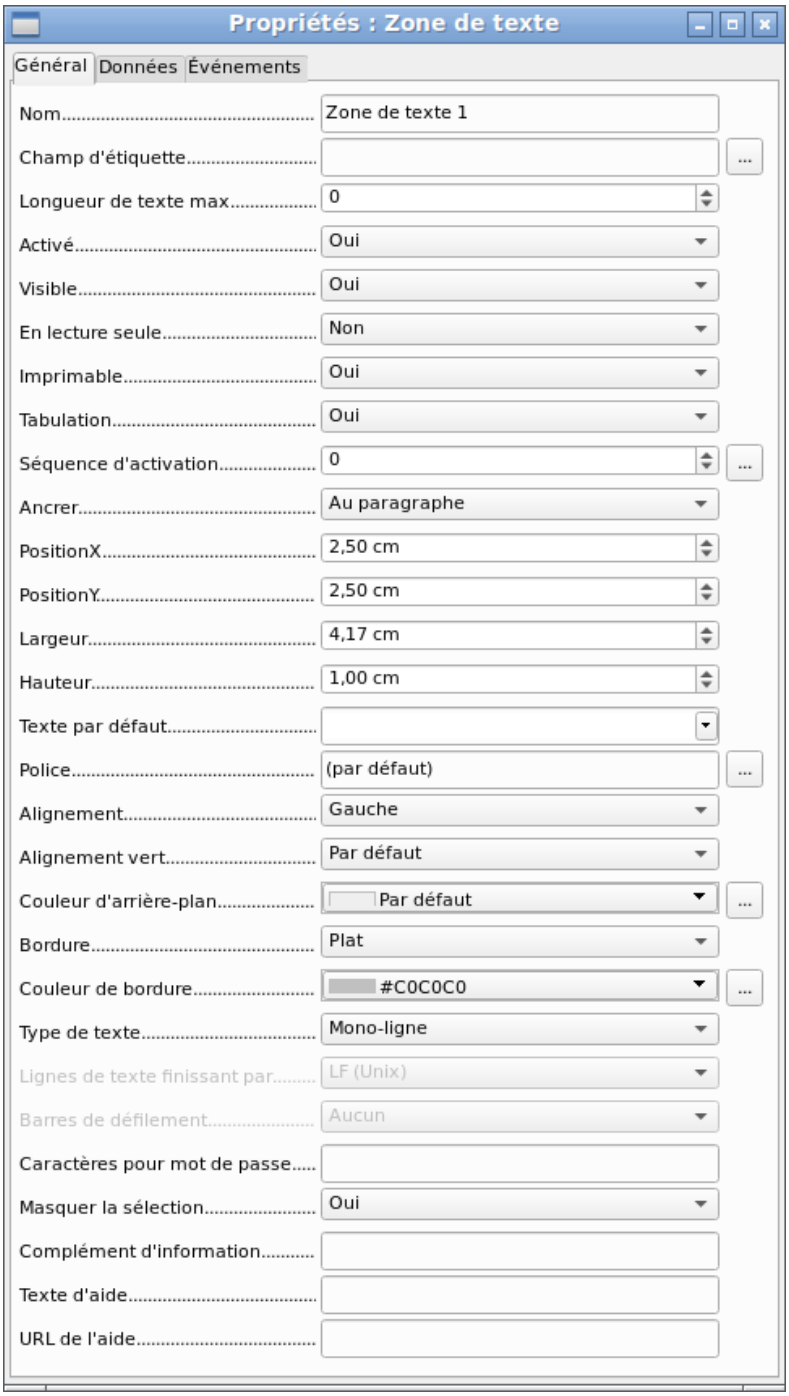


Figure 7: Propriétés de contrôle Texte

Onglet Général

Le nom d'un contrôle doit être unique dans le formulaire – utilisé pour accéder à l'aide de macros. [Name]

Le champ a-t-il une étiquette ? Cela regroupe le champ et l'étiquette.
Une étiquette permet d'accéder directement au champ du formulaire à l'aide d'un raccourci clavier.
[LabelControl]

Les champs non activés ne peuvent pas être utilisés et sont grisés. Utile pour contrôler à l'aide de macros. (Exemple : si le champ 1 contient une valeur, le champ 2 ne doit pas en contenir ; le champ 2 est désactivé.)
[Enabled]

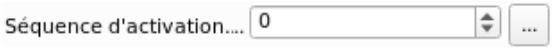
Généralement Oui ; Les champs invisibles peuvent être utilisés comme stockage intermédiaire, par exemple pour créer des champs de combinaison avec des macros. Voir le chapitre 9, Macros.
[EnableVisible]

Oui exclura toute modification de la valeur. Ceci est utile, par exemple, pour une clé primaire générée automatiquement.
[ReadOnly]

Parfois, il est utile d'imprimer une page à partir d'un formulaire plutôt que d'un rapport séparé. Dans ce cas, tous les champs ne doivent pas nécessairement apparaître.
[Printable]

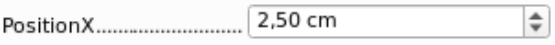
Dans un formulaire, la touche Tab est normalement utilisée pour la navigation.

Un champ en lecture seule n'a pas besoin d'un taquet de tabulation ; il peut être ignoré.
[Tabstop]

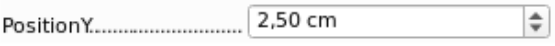


Le champ a-t-il un taquet de tabulation ? Ici, la séquence d'activation dans le formulaire est spécifiée.
[Tabindex]

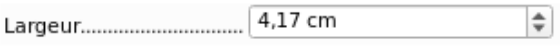
Ancrage du contrôle par rapport au texte.



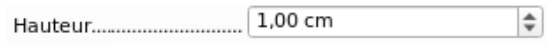
Position à partir du coin supérieur gauche relative au côté gauche du formulaire.
[PosSize.X]



Position à partir du coin supérieur gauche relative bord supérieur du formulaire.
[PosSize.Y]



Largeur du champ
[PosSize.Width]



Hauteur du champ
[PosSize.Height]

La police, la taille de la police et les effets de caractères peuvent être définis ici.
[Fontxxx]

Alignement. Ici, le de texte est justifié à gauche.
[Align]

Alignment Vertical : Par défaut | Haut | Milieu | Bas.
[VerticalAlign]

Couleur d’arrière-plan du champ de texte.
[BackgroundColor]

Encadrement : Sans cadre | Apparence3D-| plat.
[Border]

S’il y a un cadre, sa couleur peut être définie ici uniquement si Plat est sélectionné comme bordure.
[BorderColor]

Le texte en surbrillance perd la surbrillance lorsque le champ de texte perd le focus.
[HideInactiveSelection]

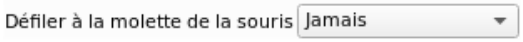
Utilisé pour les informations à lire par les macros. Voir le chapitre 9, Macros.
[Tag]

Apparaît comme une info-bulle lorsque la souris survole le champ de texte.
[HelpText]

Pointe vers un fichier d’aide, utile principalement pour HTML. Peut être appelé à l’aide de F1 lorsque le focus est sur le champ.
[HelpURL]

De plus, les champs numériques, de date, etc. ont les propriétés suivantes.

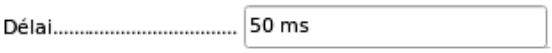
Lorsque la vérification est activée, seuls les nombres et les points décimaux peuvent être saisis.
[EnforceFormat]



N’autorise jamais les modifications à l’aide de la molette de la souris. Lorsqu’elle est sélectionnée, permet de tels changements quand le champ est sélectionné et que la souris est au-dessus du champ. Toujours signifie chaque fois que la souris survole le champ.
[MouseWheelBehavior]

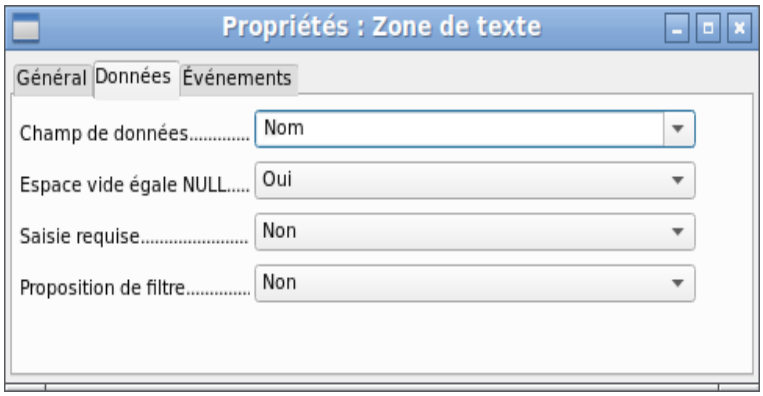
Un symbole de compteur est incorporé dans le côté droit du champ.
[Spin]

Si une flèche de rotation est enfoncée et maintenue, cela détermine si l'entrée dans la case doit être incrémentée continument au-delà de la valeur suivante.
[Repeat]



Détermine le délai minimum après une pression sur le bouton de la souris qui déclenche la répétition.
[RepeatDelay]

Onglet Données



Champ de données : Ici, vous créez la liaison avec la table, la requête ou la vue sur laquelle le formulaire est basé. [Model. DataField]

Espace vide égale NULL : Si une chaîne vide doit être traitée comme (NULL) ou le contenu simplement supprimé.

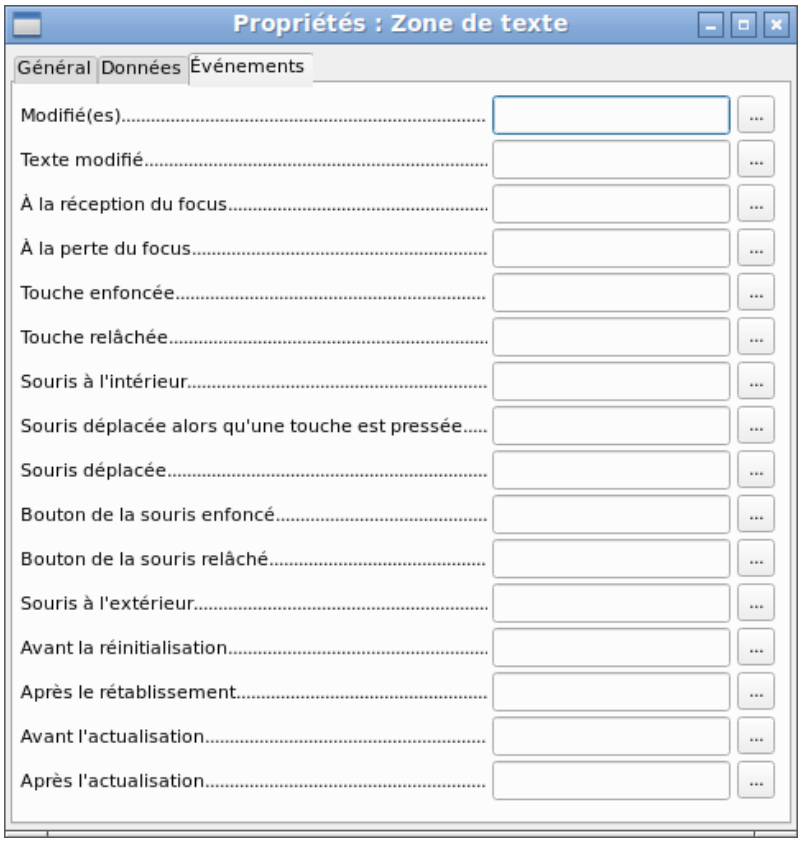
Saisie requise : Cette condition doit correspondre à celle de la table. L'interface graphique demandera une entrée si l'utilisateur n'a pas entré de valeur. [Model. InputRequired]

Proposition de filtre : Lorsque les données doivent être filtrées, le contenu de ce champ est temporairement stocké sous forme de suggestion. [Model. UserValueFilterProposal]

Attention

Attention – avec un contenu volumineux, ce choix peut nécessiter beaucoup de stockage.

Onglet Événements



Modifié(es) : Cet événement a lieu lorsqu'un contrôle est modifié et perd ensuite le focus. L'événement est perdu si vous passez directement à un autre enregistrement. Dans ces circonstances, une modification est enregistrée sans être détectée au préalable. [com.sun.star.lang. EventObject]

Texte modifié : Fait référence au contenu, qui peut en fait être du texte, des chiffres ou autre. Se produit après la saisie de chaque caractère supplémentaire. [com.sun.star.awt. TextEvent]

A la réception du focus : Le curseur entre dans le champ.

Attention

En aucun cas, la macro ne doit créer une boîte de dialogue de message à l'écran ; en cliquant dans une telle boîte de dialogue, le champ de formulaire perd le focus, puis le récupère, déclenchant à nouveau la macro. Une boucle est créée qui ne peut être interrompue qu'en utilisant le clavier.

À la perte du focus : Le curseur quitte le champ. Cela peut conduire au même type d'interaction lorsque la gestion de l'événement le fait se reproduire.

Touche : Fait référence au clavier. Par exemple, une touche est tapée lorsque vous vous déplacez dans le formulaire à l'aide de la touche Tab. Un champ reçoit le focus. Ensuite, la touche est relâchée.

L'événement est passé à l'aide du keyCode ou KeyChar de la touche libérée (lettre, chiffre, clé spéciale). [com.sun.star.awt. KeyEvent]

Souris : Explicite ; ces événements n'ont lieu que si la souris est ou était déjà dans le champ ("extérieur" correspond au javascript onMouseOut). [com.sun.star.awt. MouseEvent]

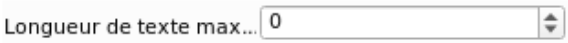
Réinitialisation/rétablissement : Le formulaire est vidé de toutes les données (lors de la création d'un nouvel enregistrement) ou remis à son état d'origine (lors de l'édition d'un enregistrement existant). Pour un champ de formulaire, cet événement est déclenché uniquement lorsque la saisie de données est annulée à l'aide du bouton dans la barre de navigation. [com.sun.star.lang. EventObject] Lorsqu'un formulaire est chargé pour la première fois, les deux événements *Avant la réinitialisation* et *Après le rétablissement* se produisent successivement, avant que le formulaire ne soit disponible pour la saisie.

Actualisation : Si l'événement est lié à un contrôle de formulaire, la mise à jour a lieu lorsque le focus est perdu et passe à un autre contrôle de formulaire, après avoir modifié le contenu du champ. Les modifications du formulaire sont acceptées et affichées. Lorsqu'un formulaire est fermé, les deux événements Avant l'actualisation et Après l'actualisation se succèdent. [com.sun.star.lang. EventObject]

Contrôle de texte

Outre les propriétés décrites à la page 14, les champs de texte peuvent avoir les propriétés supplémentaires suivantes :

Onglet Général



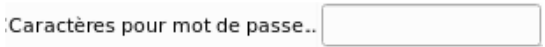
Lorsque cette valeur est 0, la saisie n'est pas autorisée. Habituellement, la longueur du champ de base de données auquel le champ de texte correspond est utilisée ici.
[MaxTextLen]

Le texte par défaut doit-il être placé dans un champ vide ? Ce texte sera supprimé si une autre saisie est effectuée avec succès.
[DefaultText]

Types possibles : Mono-ligne | Multi-ligne | Multi-ligne avec formatage (les deux derniers diffèrent par leur comportement de tabulation et, de plus, un champ de modèle ne peut pas être lié à une base de données). L'alignement vertical n'est pas actif pour les champs Multi-ligne.
[MultiLine]

Unix ou Windows ? Cela affecte principalement les fins de ligne. En interne, les lignes Windows se terminent par deux caractères de contrôle (CR et LF).
[LineEndFormat]

Uniquement pour les champs Multi-ligne : Horizontal | Vertical | Les deux.
[HScroll], [VScroll]



Actif uniquement pour les champs Mono-ligne. Modifie les caractères pour ne voir que des points.
[EchoChar]

Onglet Données

Rien de particulier.

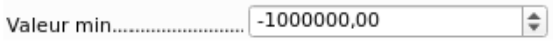
Onglet Événements

Rien de particulier.

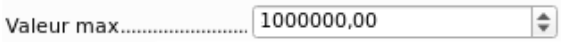
Contrôle numérique

En plus des propriétés déjà décrites, on trouve les propriétés suivantes :

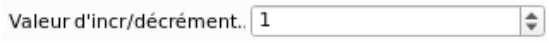
Onglet Général



Valeur minimale du champ. Doit être conforme à la valeur minimale définie dans la table.
[ValueMin]

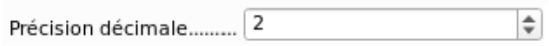


Valeur maximum.
[ValueMax]



Incrément de défilement lors de l'utilisation de la molette de la souris ou dans une boîte de sélection numérique.
[ValueStep]

Valeur affichée lors de la création d'un nouvel enregistrement.
[DefaultValue]



Nombre de décimales, 0 pour les entiers.
[DecimalAccuracy]

Séparateur pour des milliers, généralement un point.
[ShowThousandsSeparator]

Onglet Données

Il n'y a pas de contrôle pour savoir si un champ peut être NULL. S'il n'y a pas de saisie, le champ sera NULL et non 0. Aucune proposition de filtre n'est proposée.

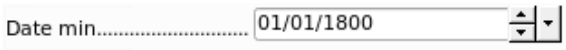
Onglet Événements

L'événement *Modifié* est absent. Les modifications doivent être gérées à l'aide de l'événement *Texte modifié* (le mot *texte* ne doit pas être pris littéralement ici).

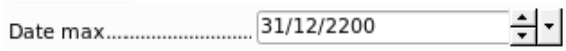
Contrôle de date

Outre les propriétés décrites à la page 14, il convient de noter les éléments suivants.

Onglet Général



Valeur minimale du champ, sélectionnable à l'aide d'un calendrier déroulant.
[DateMin]



Valeur maximum.
[DateMax]

Forme abrégée comme 27.02.20 ou formes diverses utilisant `'/'` au lieu de `'.'` ou `'-'` dans le style national/européen.
[DateFormat]

Ici, vous pouvez entrer une date littérale mais malheureusement pas (encore) la date actuelle (Aujourd'hui) au moment où le formulaire est ouvert.
[DefaultDate]

Un calendrier mensuel pour la sélection des dates peut être déroulé (Dropdown)
[DropDown]

Onglet Données

Il n'y a pas de contrôle pour savoir si un champ peut être NULL. S'il n'y a pas de saisie, le champ sera NULL et non 0.

Aucune proposition de filtre n'est offerte.

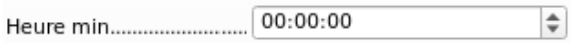
Onglet Événements

L'événement *Modifié* est absent. Les modifications doivent être gérées à l'aide de l'événement *Texte modifié* (le mot *texte* ne doit pas être pris littéralement ici).

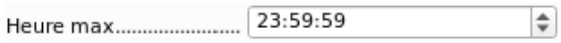
Contrôle Horaire

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles :

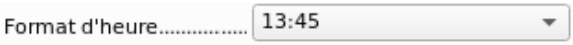
Onglet Général



Valeur minimale du champ, définie par défaut sur 0.
[TimeMin]



Valeur maximale, définie par défaut sur 1 seconde avant 24h00.
[TimeMax]



Forme courte sans secondes, forme longue avec secondes, ainsi que formats 12 heures avec AM et PM.
[TimeFormat]

Vous pouvez définir une heure fixe mais malheureusement pas (encore) l'heure réelle d'enregistrement du formulaire.
[DefaultTime]

Onglet Données

Il n'y a pas de contrôle pour savoir si un champ peut être NULL. S'il n'y a pas de saisie, le champ sera NULL et non 0.

Aucune proposition de filtre n'est offerte.

Onglet Événements

L'événement *Modifié* est absent. Les modifications doivent être gérées à l'aide de l'événement *Texte modifié* (le mot *texte* ne doit pas être pris littéralement ici).

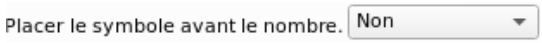
Contrôle monétaire

En plus des propriétés déjà répertoriées à la page 14, les fonctionnalités suivantes sont disponibles :

Onglet Général

Valeur min, Valeur max, Valeur d'incrément/décrément, Valeur par défaut, Précision décimale et Séparateur de milliers correspondent aux propriétés générales répertoriées à la page 21. En plus de celles-ci, il n'y a que :

Le symbole est affiché mais pas stocké dans la table qui sous-tend le formulaire.
[CurrencySymbol]



Le symbole doit-il être placé avant ou après le nombre?
[PrependCurrencySymbol]

Onglet Données

Il n'y a pas de contrôle pour savoir si un champ peut être NULL. S'il n'y a pas de saisie, le champ sera NULL et non 0.

Aucune proposition de filtre n'est offerte.

Onglet Événements

L'événement *Modifié* est absent. Les modifications doivent être gérées à l'aide de l'événement *Texte modifié* (le mot *texte* ne doit pas être pris littéralement ici).

Contrôle formaté

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles :

Onglet Général

Les valeurs minimum et maximum, ainsi que la valeur par défaut, dépendent du formatage. Derrière le bouton de mise en forme se trouve un champ flexible qui rend la plupart des champs monétaires et numériques inutiles. Contrairement à un champ monétaire simple, un modèle de champ peut afficher des sommes négatives en rouge.

Le bouton à droite avec les trois points offre un choix de formats numériques, comme vous le faites habituellement dans Calc.
[FormatKey]

Parmi les formats numériques, on peut voir, à côté de la date, de l'heure, de la devise ou du format numérique normal, des possibilités d'utilisation de champs avec une unité de mesure telle que le kg (voir Figure 8). Consultez également l'aide générale sur les codes de format numérique.

Un contrôle formaté permet de créer et d'écrire dans des champs d'horodatage de tables en utilisant un seul champ. L'Assistant Formulaire utilise une combinaison d'une date et d'un champ horaire pour cela.

Si vous souhaitez saisir des données sous la forme minutes : secondes : centièmes de secondes dans un champ d'horodatage, vous devrez utiliser des macros (voir au Chapitre 9.

Onglet Données

Rien de spécial à signaler.

Onglet Événements

L'événement *Modifié* est absent. Les modifications doivent être gérées à l'aide de l'événement *Texte modifié* (le mot *texte* ne doit pas être pris littéralement ici).

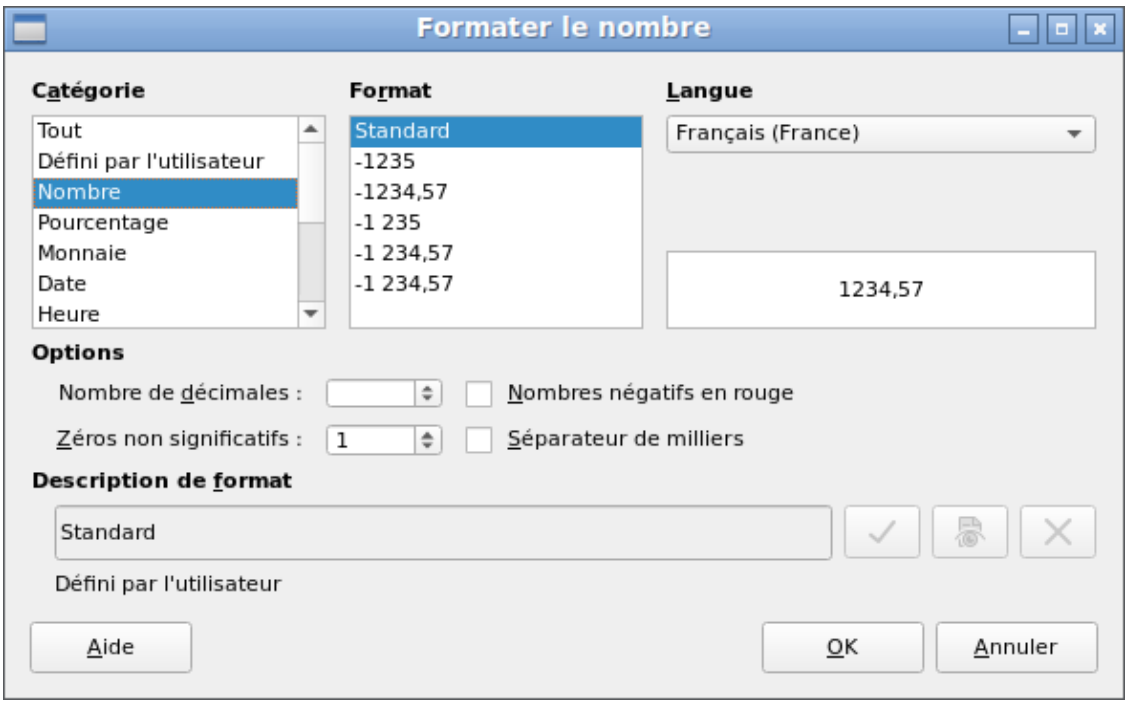
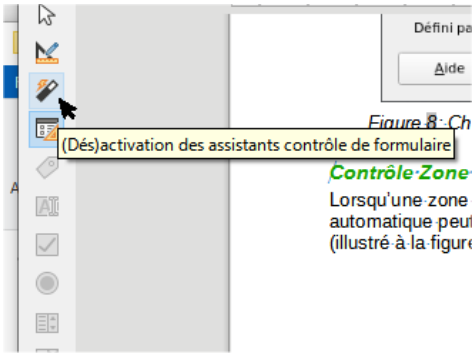
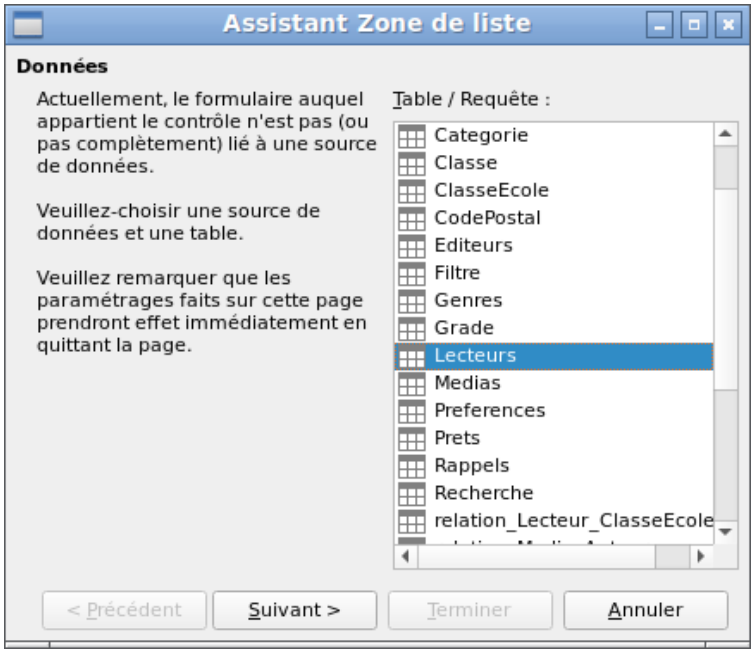


Figure 8: Champ formaté avec options numériques générales

Contrôle Zone de Liste

Lorsqu'une zone de liste est créée, l'Assistant Zone de liste apparaît par défaut. Cette apparence automatique peut être désactivée si nécessaire à l'aide du bouton **Marche / Arrêt** des assistants (illustré à la figure 1 sous l'équerre).



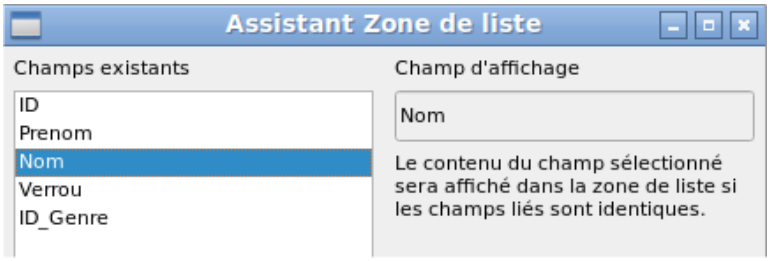


Assistant

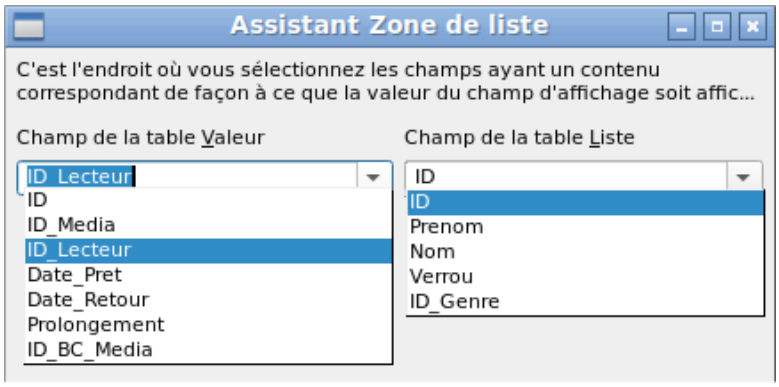
Le formulaire est déjà défini. Il est lié à une table nommée Prets. Une zone de liste montre à l'utilisateur des données différentes de ce qui est réellement transmis dans la table. Ces données proviennent généralement d'une autre table de la base de données et non de la table à laquelle le formulaire est lié.

La table des prêts est censée indiquer quel lecteur a emprunté quel média. Cependant, cette table ne stocke pas le nom du lecteur mais la clé primaire correspondante de la table Lecteurs. C'est donc la table Lecteurs qui forme la base de la zone de liste.

Le champ Nom de la table Lecteurs doit être visible dans la zone de liste. Cela sert de champ d'affichage.



Le champ ID_Lecteur apparaît dans la table Prets qui sous-tend le formulaire. Cette table est décrite ici comme la table Valeur. L'ID de clé primaire de la table Lecteurs doit être lié à ce champ. La table Lecteurs est décrite ici comme la table Liste.



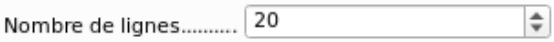
La zone de liste a maintenant été créée avec les données et la configuration par défaut. Elle est entièrement fonctionnelle.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Les entrées de la liste ont déjà été définies à l'aide de l'assistant. Sinon, vous pouvez ajouter d'autres entrées qui ne proviennent d'aucune table de la base de données. Les entrées de liste ici désignent les entrées visibles, pas celles que le formulaire transmettra à la table.
[StringItemList]

Si le champ n'est pas spécifié comme liste déroulante, des flèches de défilement apparaîtront sur le côté droit de la zone de liste lorsque le formulaire est chargé. Le champ de liste devient alors automatiquement un champ multi-ligne, dans lequel la valeur réelle sélectionnée est mise en évidence.
[Dropdown]



Si le champ est déroulant, cette propriété donne le nombre maximum de lignes visibles. Si le contenu s'étend sur plus de lignes, une barre de défilement apparaît lorsque la liste se déroule.
[LineCount]

Peut-on sélectionner plus d'une valeur ? Dans l'exemple ci-dessus, cela n'est pas possible car une clé étrangère est en cours de stockage. En général, cette fonction n'est pas utilisée pour les bases de données, car chaque champ ne doit contenir qu'une seule valeur. Si nécessaire, les macros peuvent aider à interpréter plusieurs entrées sélectionnées dans le champ de liste.
[MultiSelection] [MultiSelectionSimpleMode]

Comme le bouton désactivé l'indique clairement, une sélection par défaut n'a pas de sens dans le contexte d'une liaison avec une table de base de données, telle que créée par l'Assistant Zone de liste. Il se peut fort bien que l'enregistrement correspondant à la sélection par défaut dans l'exemple de table Lecteurs ne soit plus présent.
[DefaultSelection]

Onglet Données

En plus des propriétés de données habituelles, Champ de données et Saisie requise, il existe des propriétés importantes qui affectent la liaison entre les données affichées et les données à saisir dans la table sous-jacente au formulaire (Cf. Fig 9).

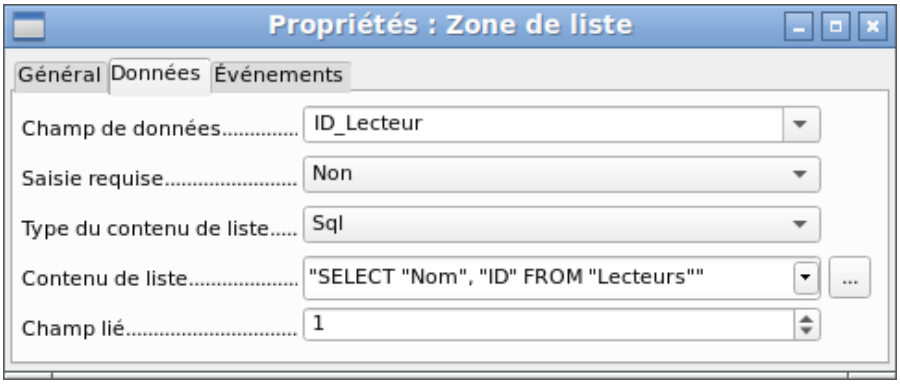


Figure 9: Onglet données d'une zone de liste

Type du contenu de liste : Liste de valeurs | Table | Requête | SQL | SQL [Natif] | Champs de table [ListSourceType]

Contenu de liste Valeurs : Si des entrées de liste ont été créées sous Général, les valeurs correspondantes à stocker sont saisies ici. Le contenu de la liste est chargé avec des éléments individuels séparés par *Maj + Entrée*. Le champ Contenu de liste les affiche alors sous la forme "Valeur1", "Valeur2", "Valeur3"… La propriété Champ lié est alors inactive.

Contenu de liste Table : Ici, l'une des tables de la base de données peut être sélectionnée. Cependant, cela est rarement possible car cela nécessite que le contenu de la table soit structuré de telle sorte que le premier champ de table contienne les valeurs à afficher dans le champ de liste, et l'un des champs suivants contient la clé primaire que la table sous-jacente au formulaire utilise comme clé étrangère. La position de ce champ dans la table est

spécifiée dans Champ lié, où *la numérotation commence par 0 pour le premier champ de la table de base de données*. Mais ce 0 est réservé à la valeur affichée, dans l'exemple ci-dessus le Nom, tandis que le 1 fait référence au champ ID.

Contenu de liste Requête : Ici, une requête est d'abord créée séparément et stockée. La création de telles requêtes est décrite au chapitre 5, Requêtes.

En utilisant la requête, il est possible de déplacer le champ ID de la première position dans la table sous-jacente à la deuxième position, ici représentée par le champ lié 1.

Contenu de liste SQL : L'Assistant Zone de liste remplit ce champ. La requête construite par l'assistant ressemble au contenu de l'option de la figure 9

La requête est la plus simple possible. Le champ Nom apparaît à la position 0, le champ ID à la position 1. Les deux sont lus à partir de la table Lecteurs. Comme le champ lié est le champ 1, cette formule SQL fonctionne. Ici devrait être ajouté *ORDER BY "Nom" ASC*. Vous n'aurez ainsi pas besoin de faire défiler trop longtemps la liste pour trouver quelqu'un. Un problème supplémentaire peut être que Nom peut être le même pour plusieurs lecteurs. Donc Prenom doit être ajouté dans la vue de la zone de liste. Lorsqu'il y a des lecteurs avec le même Nom et le même Prenom, l'ID de clé primaire doit également être affiché. Voir le Chapitre 5, Requêtes, pour plus d'informations sur son fonctionnement.

Contenu de liste SQL [Natif] : La formule SQL est saisie directement, sans utiliser l'assistant. Base n'évalue pas la requête. Cela convient lorsque la requête contient des fonctions qui peuvent ne pas être comprises par l'interface graphique de Base. Dans ce cas, la requête n'est pas vérifiée pour les erreurs éventuelles. Pour en savoir plus sur le mode SQL direct, reportez-vous au Chapitre 5, Requêtes.

Contenu de liste Champs de table : Ici, les noms de champs d'une table sont répertoriés, pas leur contenu. Pour la table Lecteurs, le contenu de la liste serait ID, Prenom, Nom, Verrou, ID_Genre.

Note

Si vous voulez un champ de temps qui peut gérer le temps en millisecondes, vous aurez besoin d'un champ d'horodatage, comme décrit dans la section sur "Contrôle Horaire". La représentation des millisecondes est incompatible avec la façon dont les caractères sont assemblés dans les zones de liste. Pour contourner ce problème, vous devez convertir l'horodatage en texte :

```
SELECT REPLACE(LEFT(RIGHT(CONVERT("Service_Requis(??)".
"Heure", VARCHAR),15),8),'.','','') AS "ContenuListe", "ID" FROM
"Service_Requis"
```

Cela donnera un affichage en minutes : secondes : centièmes.

Champ lié : Les zones de liste affichent un contenu qui n'est pas nécessairement identique à ce qui sera stocké dans le formulaire. Habituellement, un nom ou quelque chose de similaire est affiché et la clé primaire correspondante devient la valeur stockée de ce champ.

```
SELECT "Nom", "ID" FROM "Table" ORDER BY "Nom"
```

Le champ ID est stocké dans la table sous-jacente en tant que clé étrangère. Le numéro de champ dans les bases de données commence à zéro de sorte que le champ qui y est lié dans le formulaire a le numéro 1.

Si à la place vous sélectionnez "0", le contenu du champ Nom est enregistré dans le formulaire. Dans ce cas, vous pouvez supprimer le champ ID de la requête.

Il est même possible de choisir la position “–1”. Alors ce n’est pas le contenu de la requête mais la position de l’entrée qui est stockée dans la liste. Le premier enregistrement a alors la position 1.

La requête ci-dessus donne le résultat suivant :

Nom	***ID***
Anne-Lise	2
Dorothée	3
Ségolène	1

Dans les champs de liste, seul le nom peut être sélectionné. Si le champ de liste est défini sur le nom “Dorothée”, les données enregistrées suivantes sont possibles pour ce champ :

Champ lié = 1 signifie que “3” est enregistré, le contenu du champ ID.

Champ lié = 0 signifie que “Dorothée” est enregistré, le contenu du champ Nom.

Champ lié = –1 signifie que “2” est enregistré, car “Dorothée” vient en deuxième position dans la liste.

Note

Le changement du champ lié à “0” ou “–1” a été introduit avec la version 4.1 de LO. Auparavant, seule la sélection de valeurs> = 1 était possible.

Onglet Événements

En plus des événements standard, les événements suivants sont disponibles :

Exécuter l’action : Si une valeur est choisie par le clavier ou la souris, la zone de liste exécute cette action.

Statut de l’élément modifié : Cela peut être le changement du contenu affiché d’une zone de liste via l’utilisation du bouton déroulant. Cela peut également être un clic sur le bouton déroulant du champ.

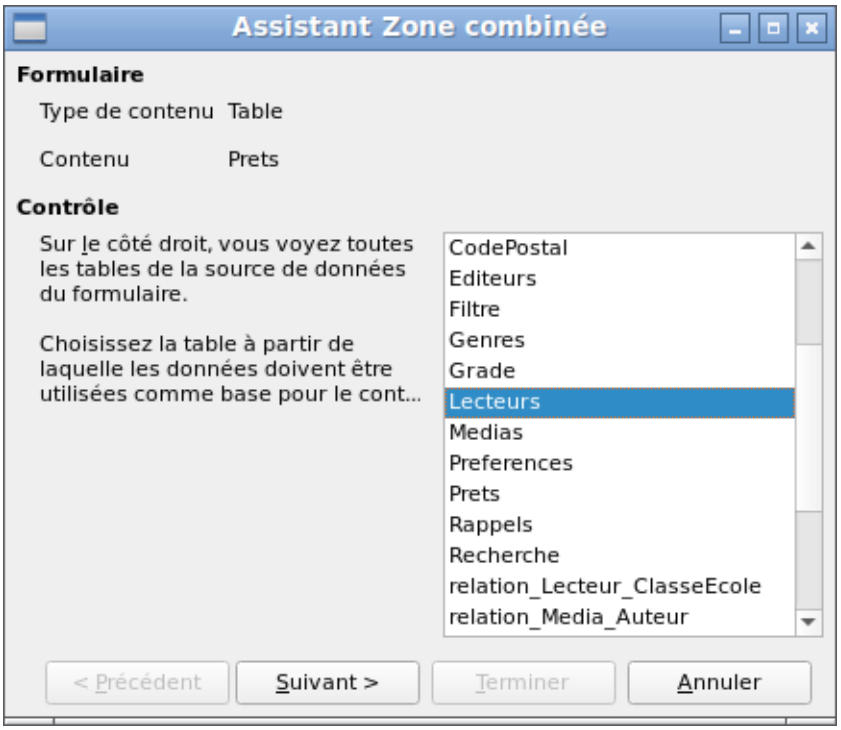
Erreur survenue : Malheureusement, cet événement ne peut pas être reconstruit pour les zones de liste.

Contrôle Zone combinée

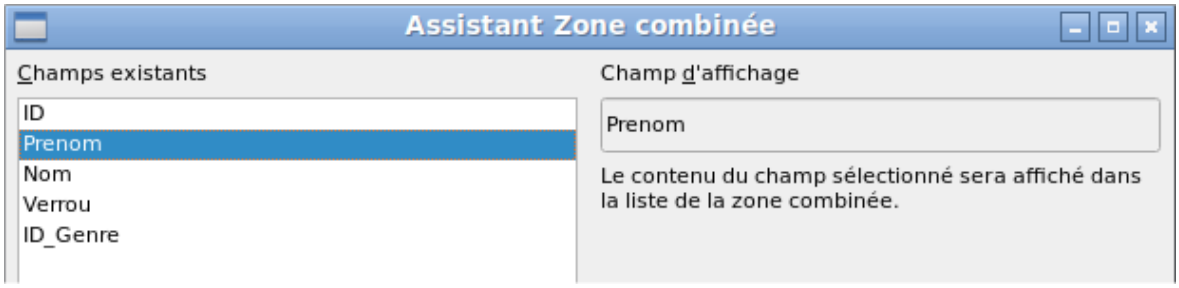
Dès qu’une zone de liste combinée est créée, un assistant apparaît par défaut, tout comme avec une zone de liste. Ce comportement automatique peut être désactivé si nécessaire à l’aide du bouton **Marche / Arrêt** des assistants.

Les zones de liste déroulante écrivent le texte sélectionné directement dans le tableau sous-jacent du formulaire. Par conséquent, l’exemple suivant montre à la fois la table liée au formulaire et celle sélectionnée pour le contrôle est la table Lecteurs.

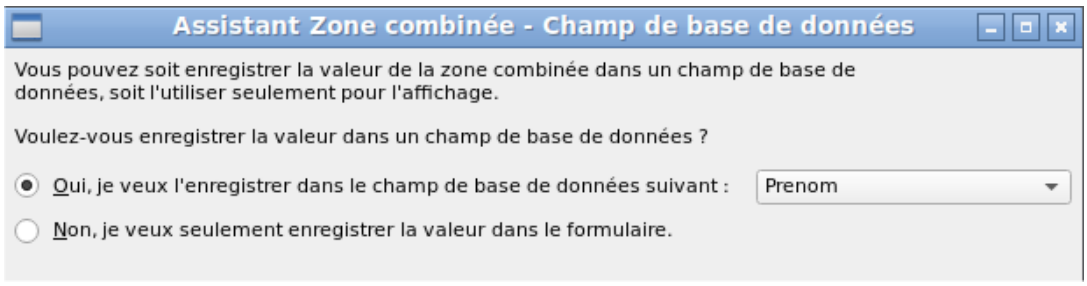
Assistant



Là encore, le formulaire est prédéfini, cette fois avec la table Lecteurs. Comme les données à afficher dans la liste déroulante doivent également être stockées dans cette table, la source sélectionnée pour les données de la liste est également la table Lecteurs.



Dans la table Lecteurs, le champ Prenom apparaît. Cela devrait être affiché dans la zone de liste déroulante.



Dans une base de données, il semble y avoir peu d'intérêt à ne pas stocker la valeur d'une zone de liste déroulante dans un champ. Nous voulons lire les prénoms de la table Lecteurs, et aussi les rendre disponibles pour les nouveaux lecteurs, afin que de nouveaux enregistrements n'aient pas besoin d'être créés pour un nom donné qui existe déjà dans la base de données. La zone de liste déroulante affiche le prénom et la saisie de texte n'est pas nécessaire.

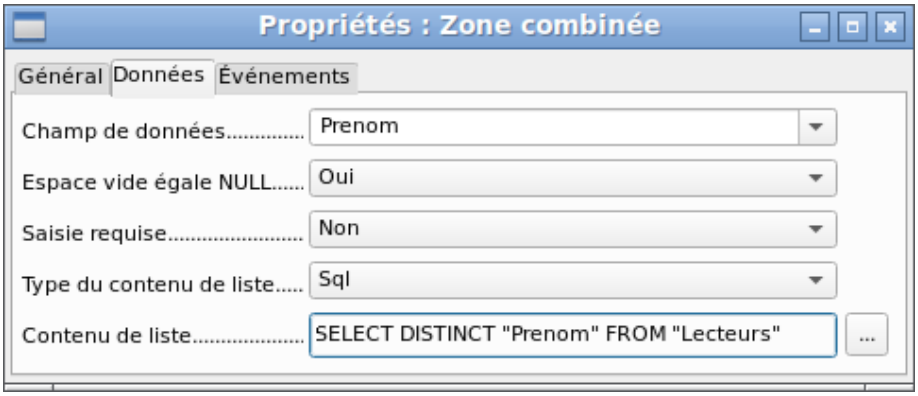
Si une nouvelle valeur doit être saisie, cela peut être fait facilement dans une zone de liste combinée, car la zone montre exactement ce qui se passe dans la table sous-jacente du formulaire.

En plus des propriétés présentées à la page 14 et décrites pour les zones de liste, les fonctionnalités suivantes sont disponibles.

Onglet Général

Lors de la saisie de nouvelles valeurs, une liste de valeurs correspondantes (le cas échéant) s'affiche pour une sélection possible.
[AutoComplete]

Onglet Données



Les champs de données sont conformes aux paramètres par défaut existants et aux paramètres d'une zone de liste. La commande SQL présente cependant une particularité :

```
SELECT DISTINCT "Prenom" FROM "Lecteurs"
```

L'ajout du mot-clé `DISTINCT` garantit que les prénoms en double ne sont affichés qu'une seule fois. Cependant, la création à l'aide de l'assistant une fois de plus rend impossible le tri du contenu.

Onglet Événements

Les événements correspondent à ceux d'une zone de liste.

Case à cocher

La case à cocher apparaît immédiatement sous la forme d'une combinaison d'un champ de case à cocher et d'une étiquette pour la case.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Le libellé de cette boîte apparaît par défaut à droite de la boîte. De plus, vous pouvez le lier à un champ d'étiquette distinct., à placer au dessus ou devant.
[Label]

Ici, en fonction du choix dans le champ *Statut triple*, jusqu'à trois possibilités sont disponibles : Non sélectionné | Sélectionné | Non défini. Non défini correspond à une entrée NULL dans la table sous-jacente au formulaire.
[État]

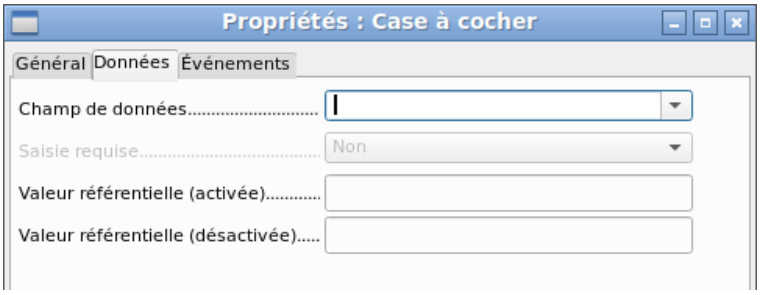
Par défaut, l'étiquette n'est pas fractionnée. L'étiquette est tronquée si le champ n'est pas assez grand.
[MultiLine]

Ici, vous pouvez spécifier une image à la place ou en plus de l'étiquette. Vous devez noter que ces images ne sont pas liées par défaut dans le document *.odb. Pour les petites images, il est utile d'incorporer l' image et non de la lier.
[Graphic]

Si vous avez choisi d'utiliser une image, son alignement avec l'étiquette peut être défini ici.
[ImagePosition] (0=gauche| 1=centré| 2=droite)

Par défaut, les cases à cocher n'ont que deux états : Sélectionné (Valeur : 1) et Non sélectionné (Valeur : 0). Avec Statut triple, une définition de champ vide (NULL) est ajoutée.
[TriState]

Onglet Données



La case à cocher peut recevoir une valeur de référence. Cependant, seules les valeurs de 1 (pour On) ou 0 (pour Off) peuvent être transférées dans le champ de données sous-jacent (les cases à cocher agissent comme des champs pour le choix de Oui et Non).

Onglet Événements

Les valeurs *Modifié, Texte modifié, Avant l'actualisation et Après l'actualisation* sont tous absents. Les champs supplémentaires pour une case à cocher sont *Exécuter l'action* (voir Zone de liste) et *Statut de l'élément modifié* (correspond à Changé).

Bouton radio

Le bouton d'option est similaire à la case à cocher décrite ci-dessus, à l'exception de ses propriétés générales et de sa forme externe (ronde).

Lorsque plusieurs Boutons d'option du formulaire sont liés au même champ de table, une seule des options peut être sélectionnée.

Onglet Général

Le bouton d'option est conçu pour être utilisé en groupe. Une seule des diverses options d'un groupe peut alors être sélectionnée. C'est pourquoi un nom de groupe apparaît ici, sous lequel les options peuvent être adressées.
[GroupName]

Onglet Données

Voir Case à cocher. Ici, cependant, les valeurs de référence saisies sont effectivement transférées dans le champ de données.

Onglet Événements

Voir Case à cocher

Contrôle picto

Un contrôle graphique (image) gère l'entrée et l'affichage des images pour la base de données. Le champ de données sous-jacent doit être un champ binaire s'il doit stocker l'image directement. Il peut également s'agir d'un champ de texte stockant le chemin relatif vers l'image. Dans ce cas, vous devez veiller à ce que le chemin d'accès aux images reste valide si la base de données est copiée.

Attention

Les images doivent dans tous les cas être réduites en taille lorsqu'elles sont stockées dans la base de données. Avec 3 Mo de photographies dans une base de données interne HSQLDB, vous obtiendrez très rapidement des erreurs Java (NullpointerException) qui rendront impossible le stockage des enregistrements. Cela peut entraîner la perte de tous les enregistrements qui ont été saisis dans la session en cours.

La saisie dans un champ image se fait soit par un double-clic avec la souris pour ouvrir une boîte de dialogue de sélection de fichier, soit par un clic droit pour choisir si un graphique existant doit être supprimé ou remplacé.

Un contrôle graphique par défaut n'a pas de taquet de tabulation.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Le graphique sélectionné ici n'est affiché qu'à l'intérieur du contrôle, pendant que le formulaire est en cours de modification. Cela n'a aucune signification pour une saisie ultérieure.
[Graphic]

Non : l'image ne sera pas adaptée au champ. S'il est trop grand, le champ s'affichera uniquement en dehors de l'image. L'image n'est pas déformée.

Conserver le rapport : l'image est adaptée au contrôle mais n'est pas déformée (rapport d'aspect conservé).

Taille Autom.: l'image est ajustée au contrôle et peut être affichée sous une forme déformée.
[ScaleImage] [ScaleMode]

Onglet Données

Rien de plus à signaler.

Onglet Événements

Les champs *Modifié, Texte modifié, Avant l'actualisation et Après l'actualisation* sont tous absents.

Contrôle de motif

Un masque de saisie est utilisé pour contrôler l'entrée dans le champ. Les caractères sont pré-spécifiés pour des positions particulières, déterminant les propriétés des caractères saisis. Les caractères prédéfinis sont enregistrés avec les caractères saisis.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Cela détermine quels caractères peuvent être saisis.
[EditMask]

C'est ce que voit l'utilisateur du formulaire.
[LiteralMask]

La longueur du masque d'édition détermine le nombre de caractères pouvant être saisis. Si l'entrée de l'utilisateur ne correspond pas au masque, l'entrée est rejetée en quittant le contrôle. Les caractères suivants sont disponibles pour définir le masque d'édition.

Caractère	***Signification***
L	Une constante de texte. Cette position ne peut pas être modifiée. Le caractère réel est affiché à la position correspondante dans le masque littéral.
a	Représente l'une des lettres a – z / A – Z. Les majuscules ne sont pas converties en minuscules.
A	Représente l'une des lettres A – Z. Si des lettres minuscules sont saisies, elles seront automatiquement converties en majuscules.
c	Représente l'un des caractères a – z / A – Z plus les chiffres 0–9. Les majuscules ne sont pas converties en minuscules.
C	Représente l'une des lettres A – Z plus les chiffres 0–9. Si des minuscules sont saisies, elles seront automatiquement converties en majuscules.
N	Seuls les chiffres de 0 à 9 peuvent être saisis.
x	Tous les caractères imprimables sont autorisés.
X	Tous les caractères imprimables sont autorisés. Si des lettres minuscules sont saisies, elles seront automatiquement converties en majuscules.

Ainsi, par exemple, vous pouvez définir le masque littéral comme “__/__/2020” et le masque d’édition comme “NNLNNLLLLL”, pour permettre à l’utilisateur de saisir quatre caractères uniquement pour une date.

Onglet Données

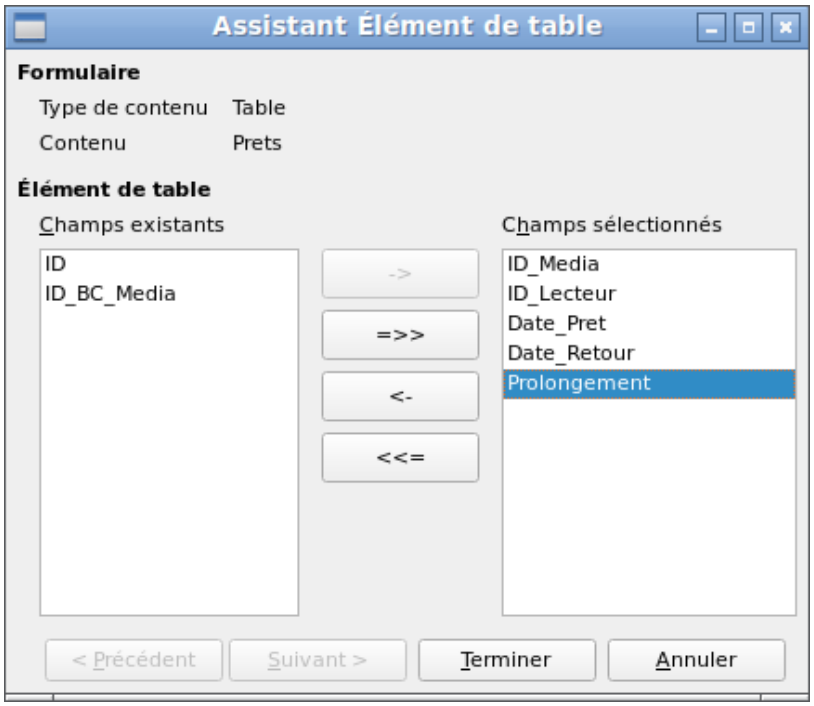
Rien de plus à signaler.

Onglet Événements

L’événement *Modifié* est absent.

Contrôle de table

Il s’agit du contrôle le plus complet. Il fournit un tableau, qui peut ensuite être doté de commandes pour les colonnes individuelles. Cela permet non seulement de visualiser les données réelles lors de la saisie, mais également les données précédemment saisies, sans qu’il soit nécessaire d’utiliser la barre de navigation pour faire défiler les enregistrements.

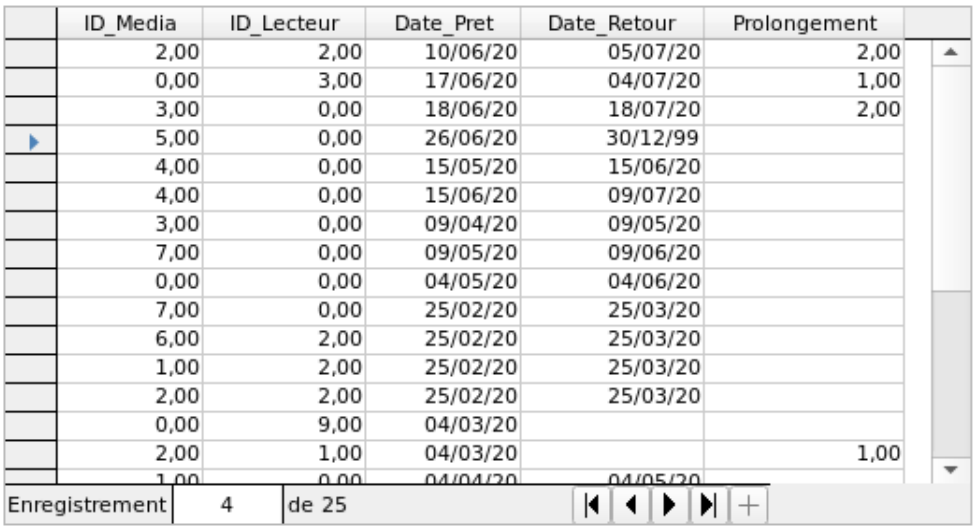


Tous les champs possibles dans un formulaire ne peuvent pas être sélectionnés pour un champ de contrôle de table. Les boutons poussoirs, les boutons d’image et les boutons d’options ne sont pas disponibles.

L’Assistant Contrôle de table assemble dans une fenêtre les champs qui apparaîtront ensuite dans le tableau.

La table Prets est disponible pour modification dans le contrôle. A part le champ ID (clé primaire) et le champ ID_BC_Media (saisie de média à l’aide d’un lecteur de code-barres), tous les champs doivent être utilisés dans le contrôle.

Le contrôle de table créé précédemment doit maintenant être développé plus avant, pour permettre l’entrée dans la table de prêts. Pour des champs tels que ID_Lecteur ou ID_Media, il serait plus utile de pouvoir choisir directement le lecteur ou le média, plutôt qu’un nombre représentant le lecteur ou le média. Pour cela, des contrôles tels que des zones de liste peuvent être placés dans le contrôle de table. Ceci est déclaré plus tard. Le formatage du champ Prolongement avec deux décimales n’était certainement pas prévu.



	ID_Media	ID_Lecteur	Date_Pret	Date_Retour	Prolongement
	2,00	2,00	10/06/20	05/07/20	2,00
	0,00	3,00	17/06/20	04/07/20	1,00
	3,00	0,00	18/06/20	18/07/20	2,00
▶	5,00	0,00	26/06/20	30/12/99	
	4,00	0,00	15/05/20	15/06/20	
	4,00	0,00	15/06/20	09/07/20	
	3,00	0,00	09/04/20	09/05/20	
	7,00	0,00	09/05/20	09/06/20	
	0,00	0,00	04/05/20	04/06/20	
	7,00	0,00	25/02/20	25/03/20	
	6,00	2,00	25/02/20	25/03/20	
	1,00	2,00	25/02/20	25/03/20	
	2,00	2,00	25/02/20	25/03/20	
	0,00	9,00	04/03/20		
	2,00	1,00	04/03/20		1,00
	1,00	0,00	04/04/20	04/05/20	

Enregistrement 4 de 25

Figure 10: Affichage du contrôle de table

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Hauteur des lignes individuelles. Sans valeur ici, la hauteur est automatiquement ajustée à la taille de la police. Les champs de texte multilignes sont alors affichés sous forme de lignes uniques à faire défiler. [RowHeight]

Comme pour les tables, le bord inférieur du contrôle affiche le numéro d'enregistrement et les aides à la navigation.[HasNavigationBar] [HasNavigationBar]

Par défaut, il y a un marqueur d'enregistrement sur le bord gauche du contrôle. Il indique l'enregistrement actuel. Vous pouvez utiliser le marqueur d'enregistrement pour accéder à la fonction de suppression de l'ensemble de l'enregistrement. [HasRecordMarker]

Onglet Données

Puisqu'il s'agit d'un champ qui ne contient pas de données mais gère d'autres champs, il n'y a pas de propriétés de données. Les données sont définies dans les colonnes.

Onglet Événements

Les événements *Modifié* et *Texte modifié* sont absents. L'événement *Erreur survenue* est ajouté.

Contrôle Étiquette

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Le titre agit comme une description d'un autre contrôle. Si le titre est lié à un contrôle, il peut être utilisé comme raccourci vers celui-ci. Un «~» inséré indique une lettre particulière qui devient le raccourci "No ~ m" définit "m" comme cette lettre. Lorsque le curseur se trouve n'importe où dans le formulaire, Alt + m va directement dans le champ.

Par défaut, une étiquette n'est pas scindée. S'il est trop long pour le champ, il est tronqué. Attention : le retour à la ligne ne reconnaît pas les espaces, donc si le champ est trop petit, une rupture peut se produire dans un mot. [MultiLine]

Onglet Données

Absent.

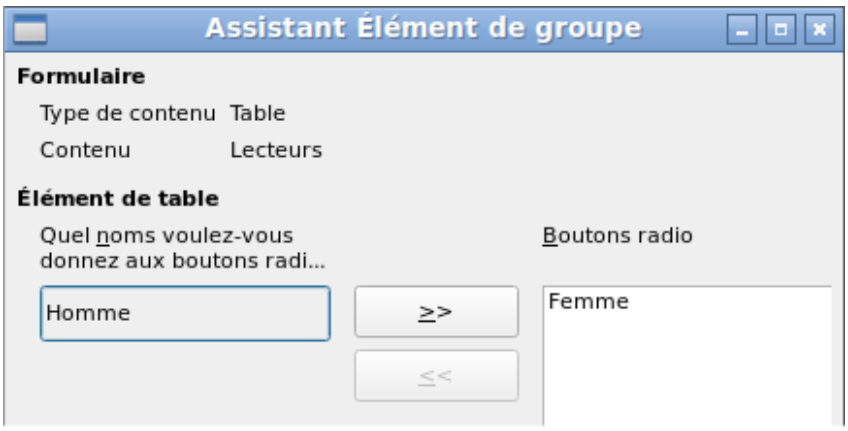
Onglet Événements

Le champ d'étiquette réagit uniquement aux événements liés à la souris, à une touche ou au focus.

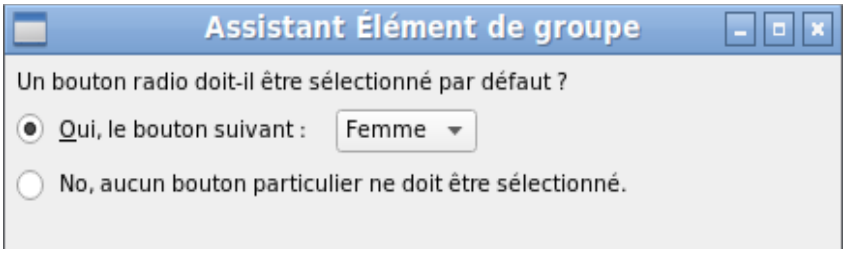
Zone de groupe

Une zone de groupe regroupe graphiquement plusieurs champs et leur fournit une étiquette collective.

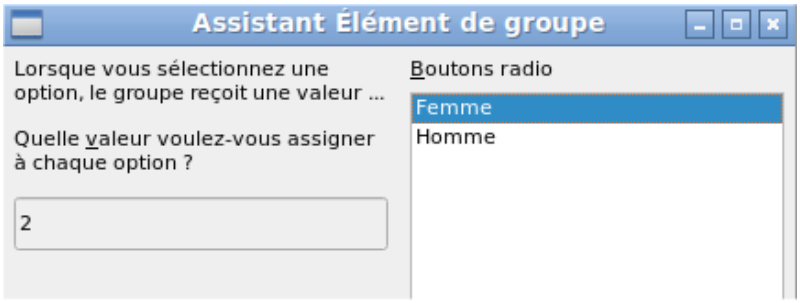
Si une zone de groupe est créée avec des Assistants actifs, l'Assistant part de l'hypothèse que plusieurs boutons d'option apparaîtront ensemble dans ce cadre.



Ce formulaire est basé sur la table Lecteurs. Nous avons affaire au choix du sexe. Les entrées sont les étiquettes des boutons d'option.

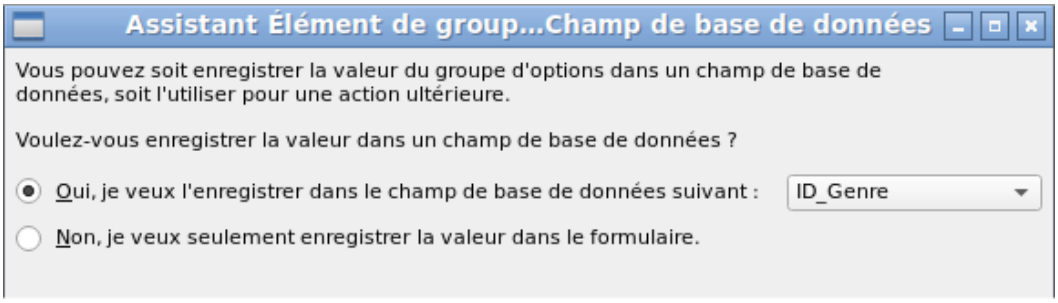


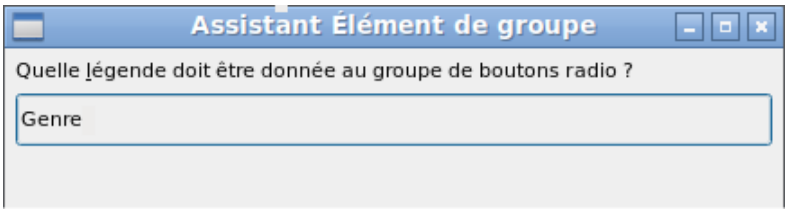
Ici, l'option par défaut est "femme". S'il n'y a pas de champ par défaut, l'entrée par défaut dans la table sous-jacente est NULL.



L'assistant donne aux boutons d'option des valeurs séparées par défaut, ici 1 pour homme et 2 pour femme. Ces valeurs correspondent aux exemples de champs de clé primaire dans la table Genres.

La valeur sélectionnée en cliquant sur un bouton d'option est transférée dans le champ ID_Genre de la table Lecteurs sous-jacente du formulaire. De cette manière, la table Lecteurs est fournie avec la clé étrangère correspondante de la table Genres en utilisant le bouton d'option.





Le groupe de boutons d'option reçoit une zone de groupe (cadre) avec l'étiquette Genre.

Si femme est sélectionnée dans le formulaire actif, l'homme est désélectionné. Il s'agit d'une caractéristique des boutons d'option qui sont liés au même champ dans la table sous-jacente. Dans l'exemple ci-dessus, les boutons d'option remplacent une zone de liste à deux éléments.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

L'étiquette peut être modifiée par rapport à sa valeur par défaut. À l'heure actuelle, les propriétés du cadre (épaisseur de ligne, couleur de ligne) ne peuvent pas être modifiées, mais vous pouvez modifier la mise en forme de la police.

Onglet Données

Aucun, car ce contrôle sert uniquement au regroupement visuel des champs.

Onglet Événements

La zone de groupe réagit aux événements impliquant la souris, une touche ou le focus.

Bouton poussoir

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

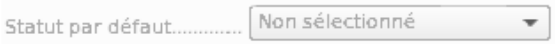
Étiquette sur le bouton.

En insérant «~», une lettre spécifique de l'étiquette peut être transformée en raccourci. "Bou~ton" définit "t" comme raccourci. Lorsque le curseur est n'importe où dans le formulaire, Alt + t vous amène au bouton.
[Label]

Le bouton reçoit le focus lorsque l'utilisateur clique dessus.

Cela n'est peut-être pas toujours souhaitable. Par exemple, vous pouvez vouloir cliquer sur le bouton pour placer du contenu dans un champ de formulaire différent, auquel cas le curseur doit rester dans ce champ lorsque vous cliquez sur le bouton.
[FocusOnClick]

Si Oui, le bouton peut être affiché enfoncé. L'état du bouton est indiqué comme pour un interrupteur. Lorsque vous appuyez dessus une deuxième fois, il affiche un bouton non enfoncé.
[Toggle]



Actif, lorsque Basculer est défini sur Oui. Sélectionné correspond au bouton enfoncé.
[DefaultState]

Habillage de mots si le bouton est trop étroit.
[MultiLine]

Une variété d'actions similaires à celles de la barre de navigation sont disponibles. Agissent sur le formulaire et la base sous jacente.

Si l'action consiste à ouvrir un document ou un site Web, placez l'URL dans le champ suivant.

HTML : Fichier à appeler avec ce bouton. Ici, vous devez choisir la ressource qui doit être ouverte par "Ouvrir le document / site Web" sous "Action".

En plus du HTML, "Action" peut être utilisé pour ouvrir d'autres modules LO. Donc si vous entrez ici.uno : RecSearch, la fonction de recherche du navigateur sera assignée au bouton.

Vous pouvez également ouvrir les fichiers Writer afin que le fait d'appuyer sur le bouton effectue un publipostage à l'aide des enregistrements de la base de données.

Les commandes disponibles ici peuvent être déterminées à l'aide de l'enregistreur de macros. [TargetURL]

Uniquement pour les formulaires HTML : cadre cible (disposition des cadres pour différentes pages HTML) dans lequel le fichier doit être ouvert. [TargetFrame]

Le bouton par défaut est encadré lorsqu'il est défini sur Oui. Lorsqu'il existe plusieurs boutons alternatifs sur un formulaire, celui qui est le plus souvent utilisé doit avoir cette caractéristique. Il est activé en appuyant sur la touche Entrée, lors qu'aucune autre action ne doit dépendre de cette touche. Un seul bouton du formulaire peut être le bouton par défaut.
[DefaultButton]

Une image doit-elle apparaître sur le bouton? [Graphic] [ImageURL]

Actif uniquement lorsqu'une image a été définie. Spécifie l'alignement du graphique sur le texte. [ImagePosition] [ImageAlign]

Onglet Données

Aucun. Un bouton effectue uniquement des actions.

Onglet Événements

Accepter l'action, *Exécuter l'action*, et *Item Statut de l'élément midifié*.

Bouton picto

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

Similaire à un bouton normal. Cependant, ce bouton n'a pas de texte et le bouton lui-même n'est pas visible. Vous ne voyez qu'un cadre autour de l'image.

Par défaut, un bouton d'image n'a pas de tabulation.

Attention : au moment de la rédaction de cet article, pratiquement aucune action ne fonctionne avec ce bouton. Il n'est pratiquement utilisable qu'avec des macros.

Onglet Données

Aucun ; ce contrôle ne réalise que des actions.

Onglet Événements

Accepter l'action et tous les événements impliquant la souris, une touche ou le focus.

Barre de navigation

Figure 11: Contrôle barre de navigation

La barre de navigation de formulaire standard est insérée dans les formulaires sur le bord inférieur de l'écran. L'insertion de cette barre d'outils peut provoquer un bref décalage vers la droite du formulaire lors de sa création à l'écran. Cela peut être gênant dans les cas où la barre de navigation est à nouveau désactivée pour certaines parties du formulaire visible, par exemple lorsqu'il existe des sous-formulaires ou plusieurs formulaires dans le formulaire visible.

En revanche, un contrôle de barre de navigation qui fait partie du formulaire, distinct des éléments correspondants, indique clairement les éléments dans lesquels vous naviguez avec la barre d'outils. Le formulaire pour les prêts, par exemple, doit d'abord rechercher parmi les lecteurs, puis montrer les médias prêtés au lecteur. Le contrôle de la barre de navigation est positionné à proximité du lecteur, de sorte que l'utilisateur remarque que la barre de navigation est utilisée pour le lecteur et non pour le média prêté au lecteur.

La barre de navigation de formulaire standard met à disposition les boutons illustrés dans la figurer 12. Le contrôle de la barre de navigation affiche les mêmes boutons à l'exception de ceux de Rechercher un enregistrement, des filtres basés sur un formulaire et une source de données sous forme de tableau.

En plus des propriétés listées à la page 14, les fonctionnalités suivantes sont disponibles.

Onglet Général

La taille de l'icône est réglable. De plus, vous pouvez choisir les groupes à afficher. Ceux-ci sont illustrées dans la figure 11 de gauche à droite en utilisant une ligne verticale comme séparateur de groupe : positionnement, navigation, action sur un enregistrement et groupes de commandes pour le filtrage et le tri.
[IconSize]
[ShowPosition]
[ShowNavigation]
[ShowRecordActions]
[ShowFilterSort]

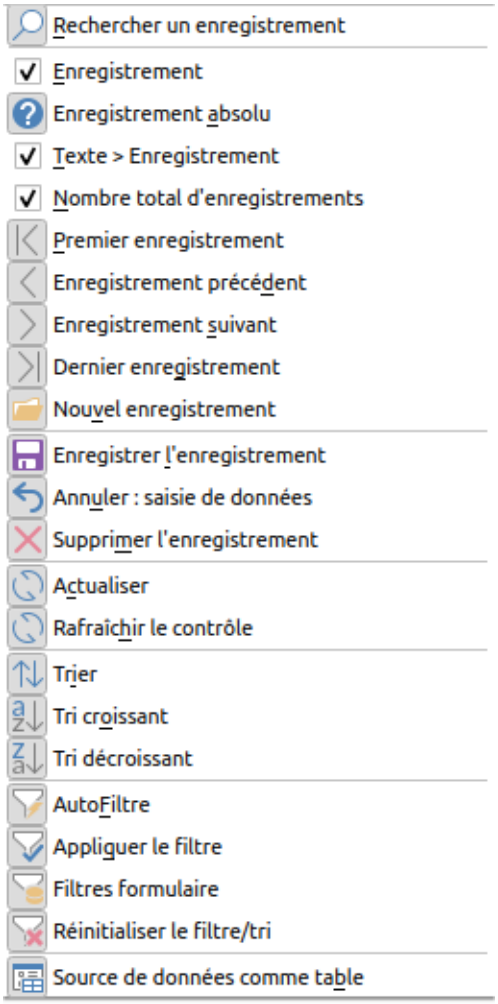


Figure 12: Boutons de navigation

Onglet Données

Aucun, car ce contrôle n'effectue que des actions.

Onglet Événements

Tous les événements qui impliquent la souris, une touche ou le focus.

Indépendamment de ce contrôle de formulaire, la barre de navigation insérable continue naturellement d'exister avec les mêmes éléments que la figure ci-dessus.

Cette barre de navigation insérable fournit en outre la recherche générale d'enregistrements, le filtre basé sur le formulaire et l'affichage de la source de données sous-jacente du formulaire en vue tableau au-dessus du formulaire.

Si vous travaillez non seulement avec un formulaire, mais avec des sous-formulaires et des formulaires auxiliaires, vous devez faire attention à ce que cette barre de navigation insérable ne disparaisse pas lorsque vous changez de formulaire. Cela crée un effet dérangeant sur l'écran.

Compteurs et barres de défilement

Aucun de ces contrôles n'a de connexion directe aux données de la base de données. Ils ne peuvent être utilisés qu'au moyen de macros, où le bouton compteur est intégré, par exemple, dans un champ numérique.

Une barre de défilement diffère d'un compteur en ayant un curseur en plus des boutons pour incrémenter et décrémenter la valeur dans une plage fixe.

Lorsque le formulaire est ouvert, la barre de défilement doit être informée de la valeur actuelle du champ auquel elle est liée. Cette macro est liée à **Propriétés du formulaire > Événements > Après le changement d'enregistrement.**

```
SUB Defilement_Formulaire(oEvent AS OBJECT)
	DIM oFormulaire AS OBJECT
	DIM oChamp AS OBJECT
	oFormulaire = oEvent.Source
	oChamp = oFormulaire.getByName("BarreDefilement")
	oChamp.ScrollValue = oFormulaire.getInt(oFormulaire.findColumn("Champ_Numerique"))
END SUB
```

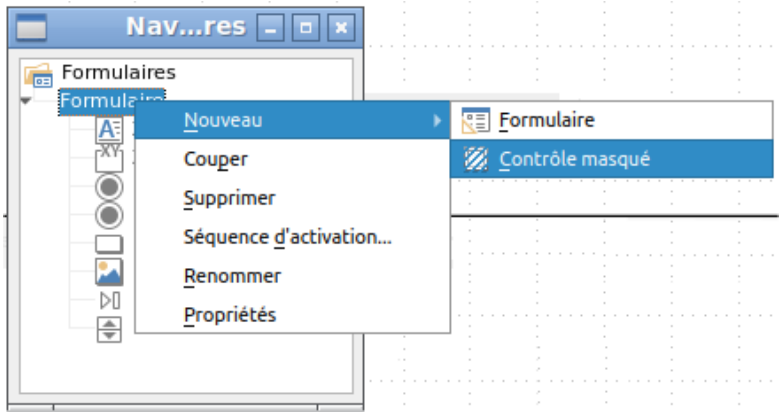
La variable est d'abord déclarée. L'événement qu'il gérera est spécifié sur le formulaire auquel la macro sera éventuellement liée. La barre de défilement sera trouvée par son nom sur le formulaire. Sa valeur actuelle est définie sur le nombre stocké dans Champ_Numerique dans la table sous-jacente. Cela peut être une clé primaire.
Si la valeur est modifiée à l'aide de la barre de défilement, cette modification doit être transmise au champ sous-jacent. Cette macro est liée à **Propriétés : Barre de défilement > Événements > A l'ajustement**.

```
SUB Defilement_Change(oEvent AS OBJECT)
	DIM oFormulaire AS OBJECT
	DIM oChamp AS OBJECT
	DIM inValeur AS INTEGER
	oChamp = oEvent.Source.Model
	inValeur = oEvent.Value
	oFormulaire = oChamp.Parent
	oFormulaire.updateInt(oFormulaire.findColumn("Champ_Numerique"), inValeur)
END SUB
```

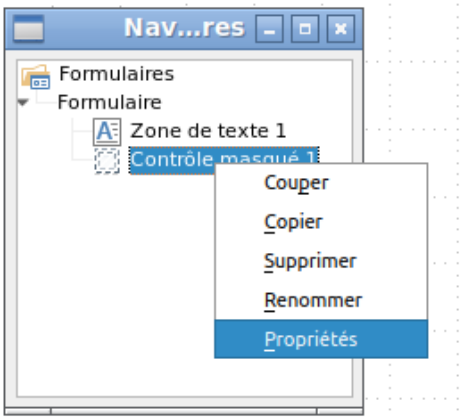
Les variables sont d'abord déclarées. La valeur sélectionnée est lue comme un entier dans le champ. Le champ correspondant dans la table sous-jacente Champ_Numerique est ensuite mis à jour à cette valeur.

Une explication détaillée de ce type de code est donnée au Chapitre 9, Macros.

Contrôle masqué



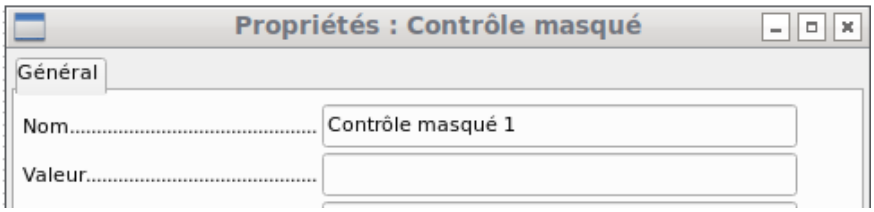
Un type de contrôle qui ne peut pas être inséré à partir de la barre d'outils contrôles est le contrôle masqué. Il doit être créé dans le navigateur de formulaire, à l'aide du menu contextuel du formulaire.



Comme tout autre contrôle, un contrôle masqué fait partie du formulaire. Cependant, il n'est pas visible dans l'interface utilisateur.

Un contrôle masqué n'a que quelques propriétés. Le nom et les informations supplémentaires ont la même signification que pour les autres contrôles. De plus, le contrôle peut se voir attribuer une valeur [Hidden Value].

Un contrôle masqué ne sert à rien si vous n'utilisez pas de macros. Cependant, avec les macros, il est souvent utile de pouvoir stocker des valeurs intermédiaires quelque part sur le formulaire, pour y accéder ultérieurement. Un exemple de cette façon d'utiliser les macros est décrit dans le chapitre 9, Macros, dans la section "Zones de liste hiérarchiques".



Sélection multiple

Si vous utilisez le bouton **Sélectionner** (flèche) pour sélectionner une grande région ou plusieurs éléments d'un formulaire, les modifications suivantes peuvent être effectuées.

Vous ne devez pas modifier le nom. Cela amènerait soudainement tous les éléments sélectionnés à acquérir le même nom. Cela rendrait difficile la recherche d'éléments individuels à l'aide du Navigateur de formulaires et la gestion du formulaire à l'aide de contrôles nommés dans les macros serait impossible.

La sélection multiple est plus utile pour changer la police, la hauteur ou la couleur d'arrière-plan des contrôles. Notez que la modification de la couleur d'arrière-plan affecte également les étiquettes.

Si vous souhaitez modifier uniquement les libellés, maintenez la touche *Contrôle* enfoncée et cliquez sur ces contrôles directement ou dans le Navigateur, ou cliquez avec le bouton droit sur un champ pour appeler les propriétés du contrôle. Désormais, le choix des propriétés que vous pouvez modifier est plus large, car vous ne traitez que des champs similaires. Vous pouvez modifier ici tout ce qui est disponible dans un champ d'étiquette.

Les possibilités de sélection multiple dépendent donc du choix des champs. Vous pouvez modifier simultanément les contrôles du même type qui ont toutes les propriétés qui existent pour une seule instance.

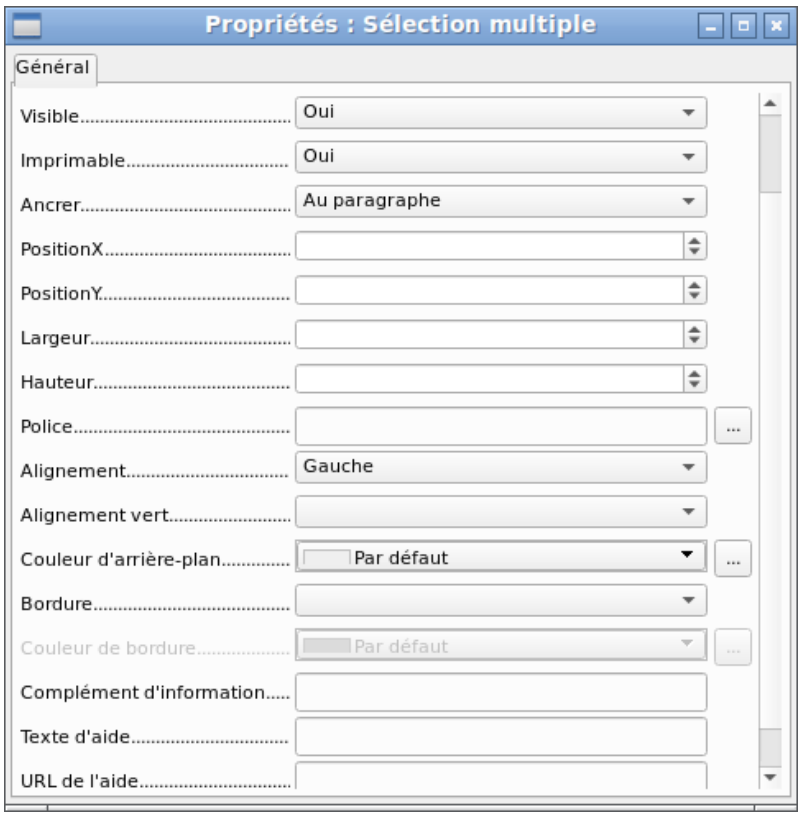
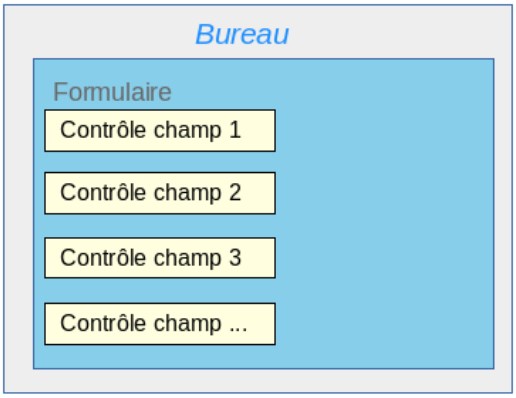


Figure 13: Propriétés générales des champs de formulaire dans une sélection multiple

Un formulaire simple rempli

Un formulaire simple a des contrôles de formulaire pour écrire ou lire des enregistrements à partir d'une seule table ou requête. Sa construction est illustrée par l'exemple suivant.

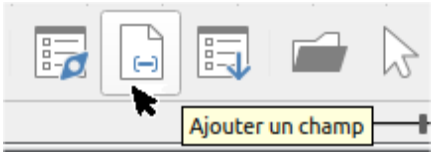


L'exemple d'un formulaire simple pour les prêts de bibliothèque est présenté ici en utilisant plusieurs variantes. La manière rapide d'utiliser l'assistant de création de formulaires est décrite au Chapitre 8, Prise en main de Base, dans le Guide de mise en route. Nous décrivons ici la création en mode Ébauche.

L’en-tête du formulaire a été créé à l’aide d’un champ d’étiquette. La police a été modifiée. Le champ d’étiquette est ancré à un paragraphe dans le coin supérieur gauche du document. À l’aide du menu contextuel du champ d’étiquette, un formulaire a été créé, lié à la table Prêts (voir“Propriétés du formulaire” en page 9). La page a également reçu un fond de couleur uniforme.

Ajout de groupes de champs

Une variante rapide pour la saisie directe de champs avec des étiquettes consiste à utiliser la fonction *Ajouter un champ*.



Cette fonction, disponible dans la barre d’outils de conception de formulaire (voir Figure 2), permet de sélectionner tous les champs de la table sous-jacente.

Double-cliquez sur les champs pour les insérer dans le formulaire en groupe avec des étiquettes (malheureusement tous au même endroit). Le groupe doit être séparé pour que le formulaire ressemble finalement à l’illustration suivante(Cf. Figure 14). Pour une meilleure vue, toutes les barres d’outils inutiles ont été supprimées de la fenêtre, qui a également été compressée afin que tous les éléments de la barre de navigation ne soient pas visibles.

Tous les champs ont été sélectionnés à l’exception de ID_BC_Media, qui est conçu pour être utilisé uniquement avec un lecteur de codes-barres.

Conseil

Juste après avoir ajouté un champ, le déplacer sur la feuille (vers sa position définitive, si possible), pour éviter la superposition avec ceux créés ultérieurement ou le sélectionner et le glisser dans le formulaire.

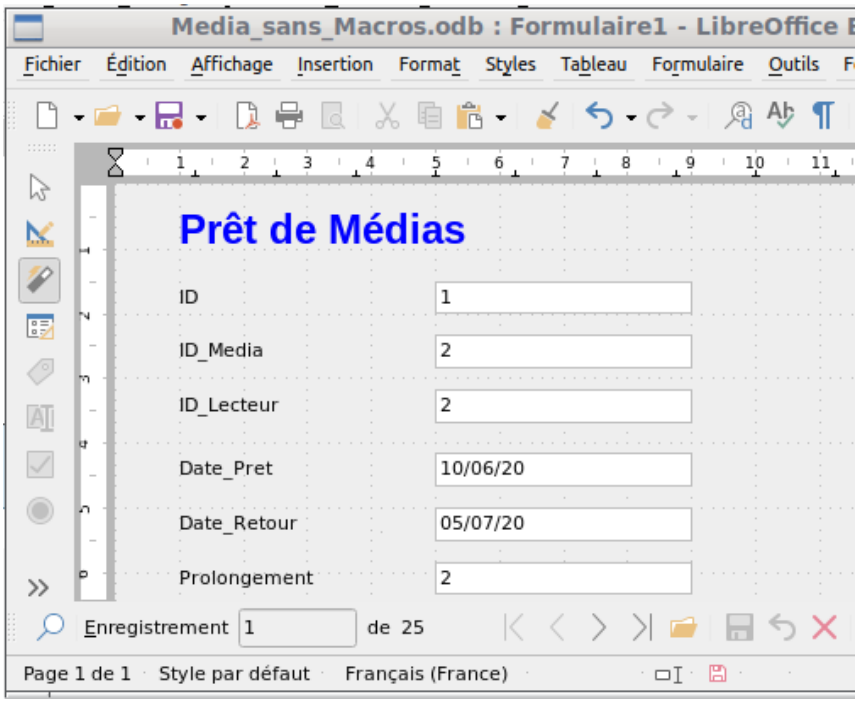


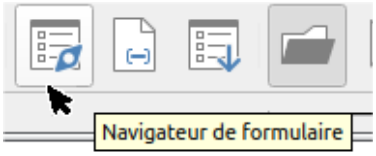
Figure 14: Formulaire simple créé en utilisant Ajouter un champ

Pour chaque champ de la table, un contrôle de formulaire approprié a été automatiquement sélectionné. Les nombres sont dans des champs numériques et sont déclarés comme des entiers sans décimales. Les champs de date sont correctement représentés en tant que contrôles de date. Tous les champs ont la même largeur. Si un contrôle graphique avait été inclus, il aurait reçu un champ carré. Voir l'exemple *Form01-VoirPrets de l'exemple Media_sans_Macros.*

Ajustement des proportions de champ

Nous pouvons maintenant faire des choses créatives, y compris ajuster la longueur des champs et transformer les dates en champs déroulants. Il est encore plus important de rendre les champs ID_Media et ID_Lecteur plus conviviaux, à moins que chaque utilisateur de bibliothèque ne dispose d'une carte d'identité de bibliothèque et que chaque support soit fourni avec un ID lors de l'adhésion. Cela ne sera pas supposé dans ce qui suit.

Pour ajuster les champs individuels, nous devons éditer le groupe. Cela peut être fait avec un clic droit sur le groupe puis en suivant le menu contextuel (Figure 15). Pour les travaux futurs, ce sera plus clair si nous utilisons le Navigateur de formulaires.



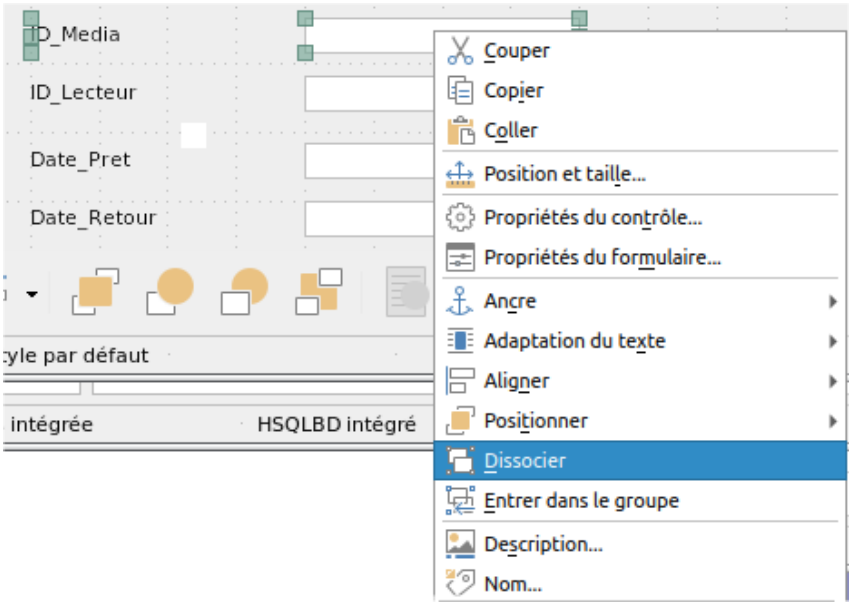


Figure 15: Contrôles : Modification du groupe Contrôle/Etiquette

Le navigateur de formulaire affiche tous les éléments du formulaire avec leurs étiquettes. Pour les contrôles, les noms proviennent directement des noms des champs de la table sous-jacente. Les noms des étiquettes ont le préfixe Étiquette.

Un clic sur ID_Media sélectionne ce champ (Figure 16). Cliquez avec le bouton droit de la souris pour remplacer le champ sélectionné par un autre type de champ, à l'aide du menu contextuel (Figure 17).

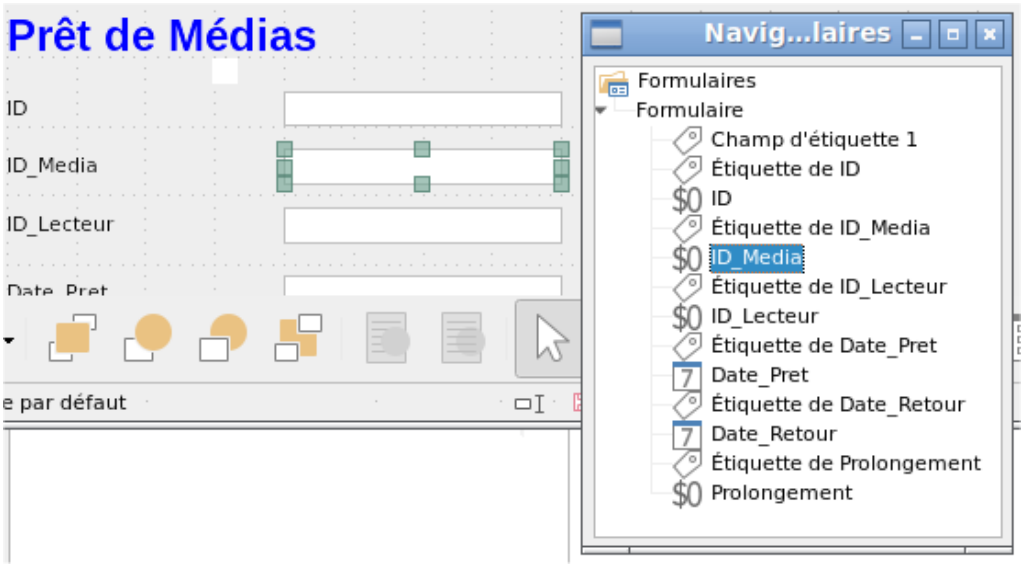


Figure 16: Sélection des contrôles de formulaire directement à l'aide du navigateur de formulaires

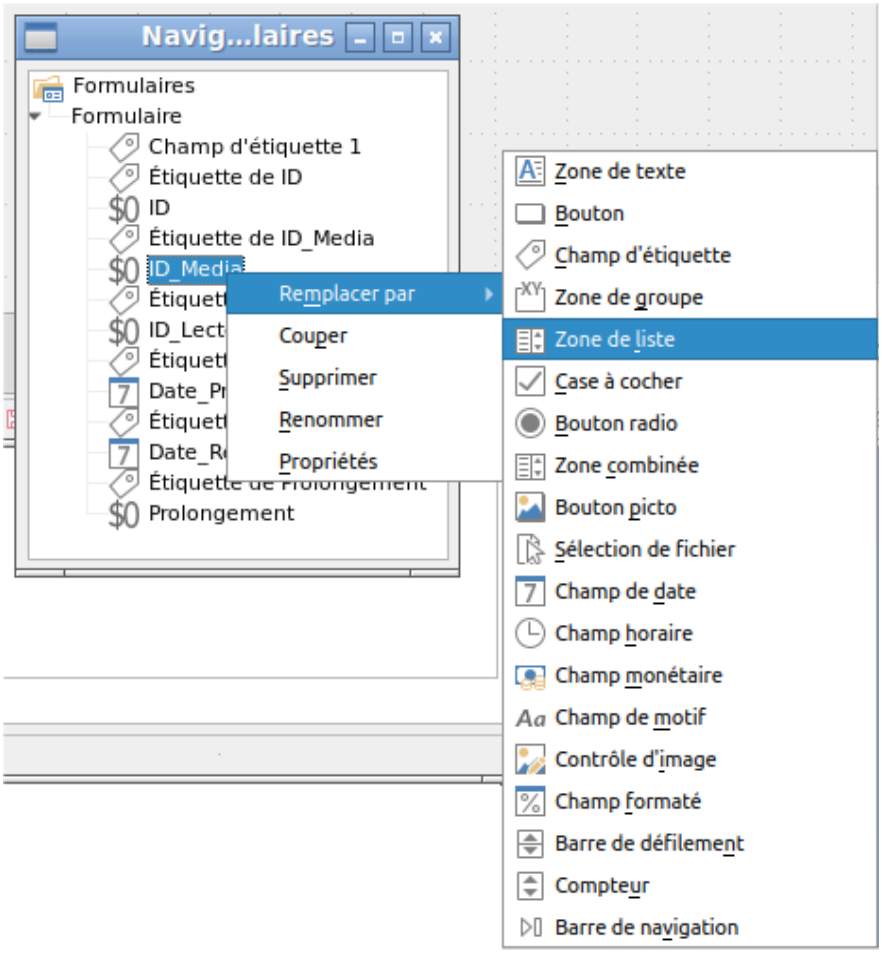
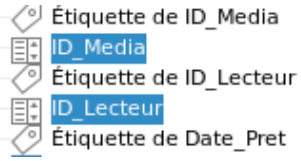


Figure 17: Remplacement d'un type de contrôle par un autre à l'aide du navigateur de formulaires

Ce remplacement est effectué pour les contrôles ID_Media et ID_Lecteur.



La modification est rendue visible dans le Navigateur de formulaires par la modification de l'icône associée.

La requête SQL pour le champ de liste peut maintenant être créée via l'interface utilisateur graphique en cliquant sur le bouton à droite.

Ceci est effectué automatiquement lorsqu'une zone de liste est créée directement, mais pas lorsqu'elle est formée par conversion à partir d'un autre type de contrôle.

Les contrôles Zone de liste permettent d'afficher autre chose que les champs ID sans utiliser de macro.

Les requêtes :

- Pour le champ ID_Media (étiquette "Media")
  ```
  SELECT "Titre", "ID" FROM "Medias"
  ```
- Pour le champ ID_Lecteur (étiquette "Lecteur")
  ```
  SELECT "Nom" || ', ' || "Prenom","ID" FROM "Lecteurs"
  ```

Pour plus de détails sur la commande SQL, reportez-vous au Chapitre 5, Requêtes. Voir l'exemple *Form02-VoirPretsListe de l'exemple Media_sans_Macros.*

Figure 18: Remplacement par zones de liste et requêtes SQL

Étant donné que les listes déroulantes doivent être créées, les défauts suivants peuvent être corrigés en même temps :

- Les étiquettes des zones de liste doivent être Media au lieu de ID_Media et Lecteur au lieu de ID_Lecteur.
- Le contrôle ID doit être déclaré en lecture seule.
- Les champs qui ne sont pas absolument nécessaires pour l'octroi de prêts pour un nouveau support n'ont pas besoin d'un taquet de tabulation. Sans cela, le formulaire peut être parcouru beaucoup plus rapidement. Si nécessaire, la tabulation peut également être ajustée à l'aide de la séquence d'activation (voir page 14). Seuls les champs Media, Lecteur et Date_Pret doivent être accessibles dans tous les cas à l'aide de la touche Tab.
- Si le formulaire est destiné à la réalisation de prêts, il est inutile et également déroutant pour les médias retournés d'être affichés. Les supports avec une date de retour doivent être filtrés. De plus, l'ordre d'affichage pourrait être trié par lecteur, de sorte que les médias prêtés à la même personne soient affichés successivement. Voir la note sur"Propriétés du formulaire" en page 9. Cependant, il y a un problème ici en ce que les lecteurs peuvent être triés uniquement par ID, pas par ordre alphabétique, car la table sous-jacente au formulaire contient uniquement l'ID.

Ajout de champs uniques

L'ajout de champs uniques est un peu plus compliqué. Les champs doivent être sélectionnés, glissés sur la surface du formulaire et le champ approprié de la table sous-jacente spécifié. De plus, le type de champ doit être correctement choisi ; par exemple, les champs numériques ont deux décimales par défaut.

Ce n'est que lors de la création de zones de liste que l'assistant entre en jeu, ce qui permet à un novice d'effectuer plus facilement les étapes de création des champs appropriés, jusqu'à un certain point. Au-delà de ce point, l'assistant cesse de répondre aux exigences car :

- Les entrées ne sont pas triées automatiquement.
- La combinaison de plusieurs champs dans le contenu de la zone de liste n'est pas possible.

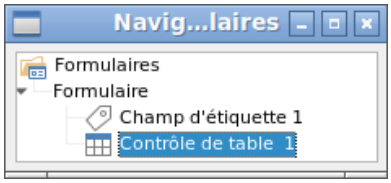
Ici encore, nous devons apporter des améliorations rétrospectives, afin que le code SQL requis puisse être créé assez rapidement à l'aide de l'éditeur de requête intégré.

Lors de l'ajout de champs uniques, le champ et son libellé doivent être explicitement associés (voir "Paramètres par défaut pour de nombreux contrôles" en page 14). En pratique, il pourrait être préférable de ne pas associer de champs aux étiquettes, de sorte que vous n'ayez pas à utiliser la modification de groupe avant de modifier les propriétés d'un champ. Cela dit, Un champ d'une association Champ-étiquette est toujours accessible par ctrl-clic sur ce champ, ce qui facilite les modifications

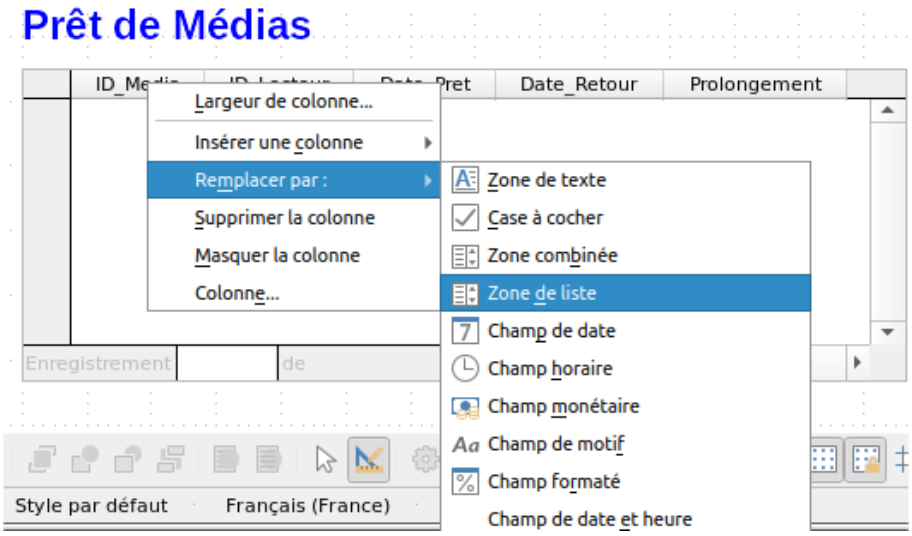
Contrôle de table

L'utilisation de l'assistant de table pour créer un champ de table a déjà été décrite à la page 34. Il présente cependant quelques défauts qui doivent être améliorés :

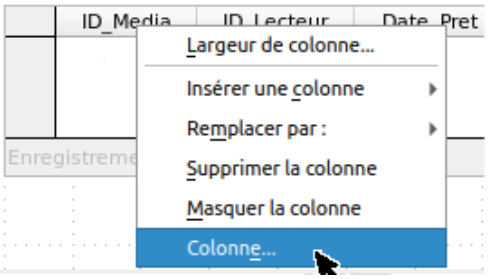
- Les champs ID_Media et ID_Lecteur doivent devenir des zones de liste.
- Les champs numériques doivent être débarrassés de leurs décimales, car l'assistant spécifie toujours deux décimales pour les nombres.



La modification des champs dans le contrôle de table n'est pas possible en utilisant la même méthode que celle décrite pour les autres contrôles. Dans le navigateur, la description des champs se termine par le champ Table. Le Navigateur ne sait rien des contrôles qui se trouvent dans le contrôle de table, faisant référence aux champs de la table sous-jacente. Cela s'applique également plus tard, lorsque des tentatives sont faites pour accéder aux champs à l'aide de macros. Ils ne sont pas accessibles par leur nom.



Les contrôles dans le contrôle de table sont appelés colonnes. Grâce au menu contextuel, il est désormais possible de remplacer un type de champ par un autre. Cependant, toute la gamme des types n'est pas disponible. Il n'y a pas de boutons poussoirs, de boîtes d'options ou de commandes graphiques.



Les propriétés des champs sont masquées dans le menu contextuel derrière le concept de colonnes. Ici, par exemple, le champ numérique Prolongement peut être modifié afin qu'aucune décimale ne soit affichée. De plus, la valeur minimale par défaut de –1 000 000,00 n'a guère de sens pour une prolongation de prêt. Le nombre doit toujours rester petit et positif.

Dès que les propriétés d'une colonne sont affichées, vous pouvez sélectionner une autre colonne sans fermer la boîte de dialogue des propriétés. De cette façon, vous pouvez travailler sur tous les champs, l'un après l'autre, sans avoir à enregistrer entre les deux.

Note

La valeur est enregistrée dans la boîte de dialogue dès qu'un mouvement est effectué entre les propriétés. Tel est également le cas lors du passage d'une colonne à la suivante.

Terminez en sauvegardant le formulaire entier, et enfin la base de données elle-même.

Les propriétés des champs intégrés dans un contrôle de table ne sont pas aussi complètes que celles des champs individuels . La police, par exemple, ne peut être définie que pour l'ensemble du contrôle de table. De plus, vous n'avez pas la possibilité de sauter des colonnes individuelles en supprimant leurs taquets de tabulation.

Conseil

Vous pouvez vous déplacer dans un formulaire à l'aide de la souris ou de la touche Tab. Si vous tabulez dans un contrôle de table, le curseur déplacera un champ vers la droite pour chaque tabulation supplémentaire. À la fin de la ligne, il reviendra au premier champ de l'enregistrement suivant dans le contrôle de table. Pour quitter le contrôle de table, utilisez Ctrl + Tab.

Si seuls certains champs doivent être visibles pendant l'utilisation, vous pouvez utiliser plusieurs contrôles de table différents dans le formulaire, car l'onglet est capturé par défaut par chaque contrôle de table.

Le formulaire illustré à la figure 19 concerne le prêt de médias. Seuls les champs immédiatement nécessaires sont affichés dans le champ de table supérieur. La partie inférieure montre tous les champs, de sorte qu'il soit évident à quelle personne et à quel support le retour est destiné.

Note

Les colonnes ID_Media et ID_Lecteur ont été remplacées par des zones de liste, et leur zone de données remplacées par des requêtes SQL identiques à celles de la page 49. Les étiquettes des colonnes ont été modifiées en conséquence.

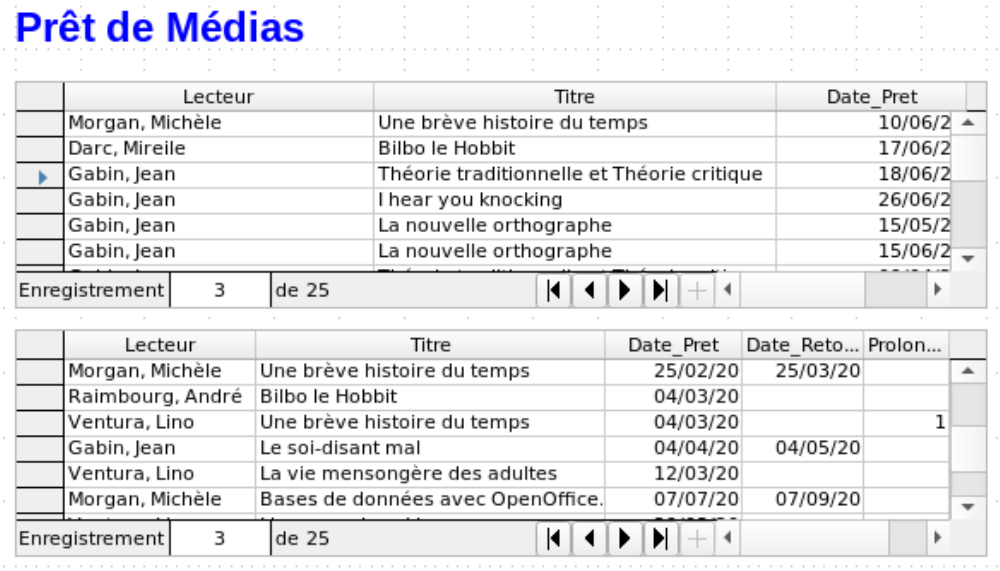
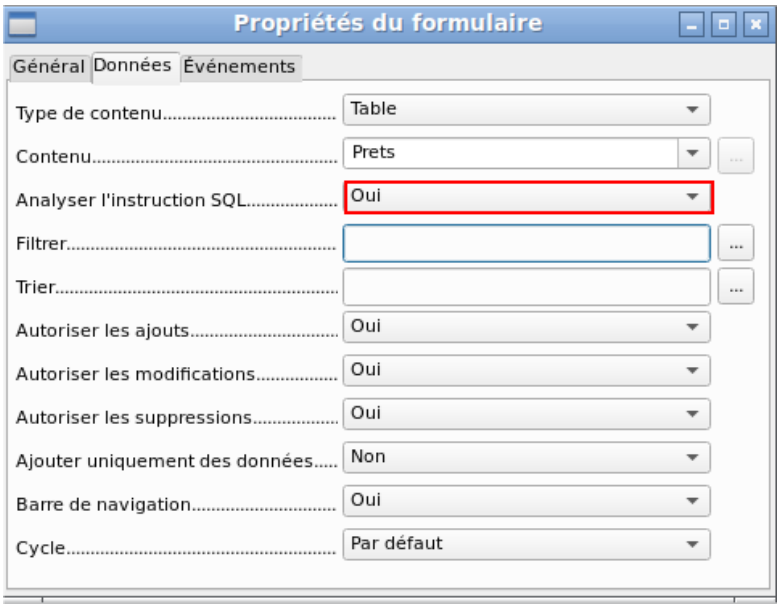


Figure 19: Un formulaire avec plusieurs contrôles de table

Cette figure montre un défaut esthétique qui nécessite une attention particulière. Dans le contrôle de table supérieur, le même support apparaît parfois plus d'une fois. Cela se produit, car le tableau affiche également les médias qui ont été rendus précédemment. Par conséquent, les données doivent être filtrées pour afficher uniquement les prêts. Les enregistrements avec une date de retour ne doivent pas apparaître.

Ce filtrage peut être réalisé soit par requête, soit directement à l'aide des propriétés du formulaire. Si cela est fait à l'aide des propriétés du formulaire, le filtre peut être temporairement désactivé pendant la saisie. Le filtrage à l'aide d'une requête est décrit au chapitre 5, Requêtes. Nous décrivons ici comment le faire à l'aide des propriétés du formulaire.

Pour avoir accès au Filtre, il faudra valider *Analyser l'instruction SQL*.



Le filtrage est effectué à l'aide du bouton aux trois points (une ellipse, qui ouvre la boîte de dialogue ci-dessous). Vous pouvez également saisir le filtre directement dans le champ de texte Filtre si vous connaissez le codage SQL.

À l'aide de l'interface graphique, vous pouvez maintenant sélectionner le champ nommé Date_Retour. Il affichera uniquement les enregistrements pour lesquels le champ est vide, où "vide" correspond à la désignation SQL NULL.

Le formulaire nettoyé (illustré à la figure 20) semble maintenant plus simple.

Bien sûr, il y a encore place à l'amélioration, mais par rapport à la forme précédente, cette version présente un net avantage en ce que tous les médias sont visibles d'un coup d'œil.

Le traitement des données à l'aide de contrôles de table est similaire à l'utilisation d'une table réelle. Un clic droit sur l'en-tête de l'enregistrement d'un enregistrement existant entraîne sa suppression, et une entrée peut être annulée ou enregistrée dans le cas de nouveaux enregistrements.

Lorsque vous quittez une ligne, l'enregistrement est automatiquement sauvegardé.

Figure 20: Formulaire modifié avec le filtre

Nous pouvons encore améliorer le formulaire de prêt de médias de plusieurs façons.

- Ce serait bien si la sélection d'un lecteur dans une partie du formulaire provoquait l'affichage du média prêté à ce lecteur dans une autre.
- Dans le tableau ci-dessus, vous pouvez voir un grand nombre d'enregistrements qui ne sont pas nécessaires, car ces supports sont déjà prêtés. Le tableau a été créé pour

permettre l'octroi de prêts, il serait donc préférable que seule une page vide apparaisse, qui pourrait ensuite être remplie avec le nouveau prêt.

De telles solutions sont disponibles en utilisant d'autres formulaires qui sont organisés de manière hiérarchique et permettent des vues séparées des données. Voir l'exemple *Form03-VoirPretsFiltre*

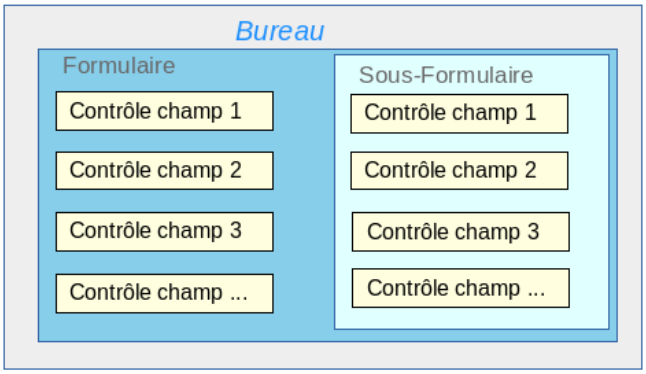
Conseil

Les contrôles de table peuvent être combinés avec d'autres champs du formulaire. Par exemple, le champ de contrôle de table affiche uniquement les titres des supports pour lesquels des modifications ont été apportées dans les champs de formulaire sous-jacents.

Lorsque les contrôles de table sont combinés avec d'autres champs de formulaire, il existe un petit bogue, qui peut être facilement contourné. Si les deux types de champ sont présents ensemble dans un formulaire, le curseur se déplace automatiquement des autres champs dans le contrôle de table même si par défaut ce type de champ a le paramètre **Tabulation > Non**. Cela peut être résolu en basculant la propriété Tabulation sur Oui, puis de nouveau à Non. Cela entraînera l'enregistrement de la valeur "Non".

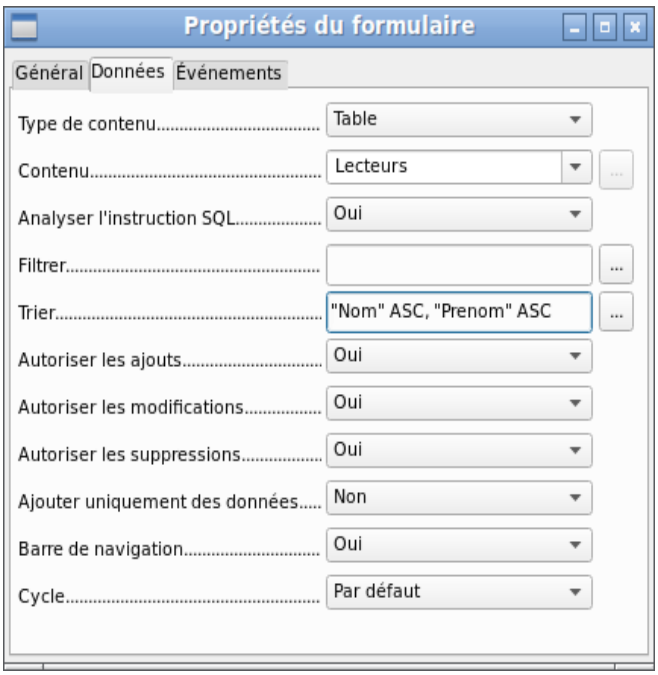
Formulaires principaux et sous-formulaires

Un sous-formulaire se trouve dans un formulaire comme un contrôle de formulaire. Il est lié aux données du formulaire principal. Cependant sa source de données peut être une autre table ou une requête (ou une commande SQL). La chose importante pour un sous-formulaire est que sa source de données est en quelque sorte liée à la source de données du formulaire principal.

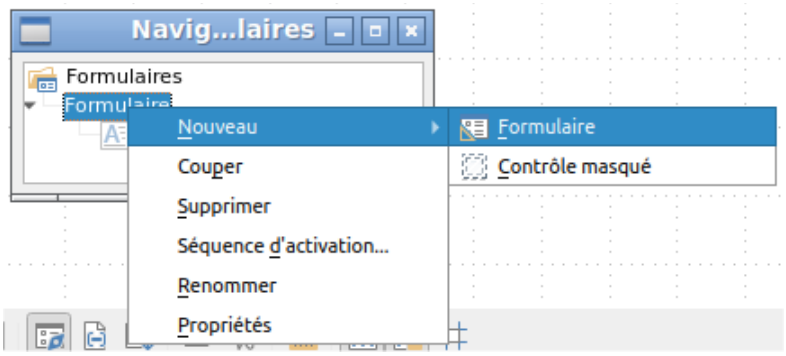


Les structures de table typiques qui se prêtent à être utilisées comme sous-formulaires sont les tables avec une relation un-à-plusieurs (voir Chapitre 3, Tables). Le formulaire principal affiche une table avec des enregistrements auxquels de nombreux enregistrements dépendants du sous-formulaire peuvent être liés et affichés.

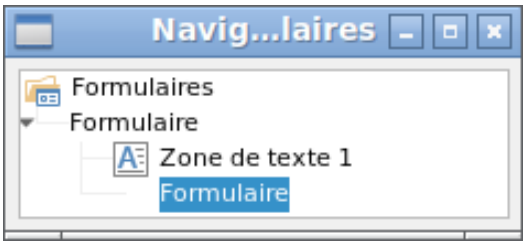
Nous allons d'abord utiliser la relation entre la table Lecteurs et la table Pret (voir Chapitre 3, Tables). La table Lecteurs constituera la base du formulaire principal et la table des Prets sera reproduite dans le sous-formulaire.

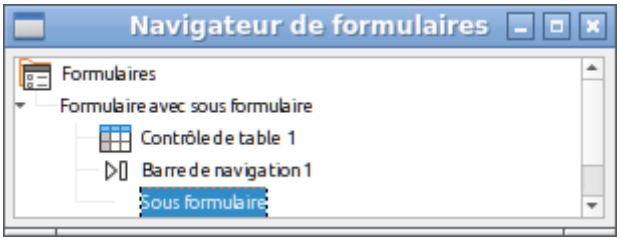


Ici, le formulaire principal est lié à la table Lecteurs. Pour accélérer la recherche de lecteurs, le tableau est trié par ordre alphabétique. Nous nous passerons d'une barre de navigation, puisque le contenu du sous-formulaire se situerait entre le formulaire principal et la barre de navigation. À la place, nous utiliserons le contrôle de formulaire intégré (Figure 11).

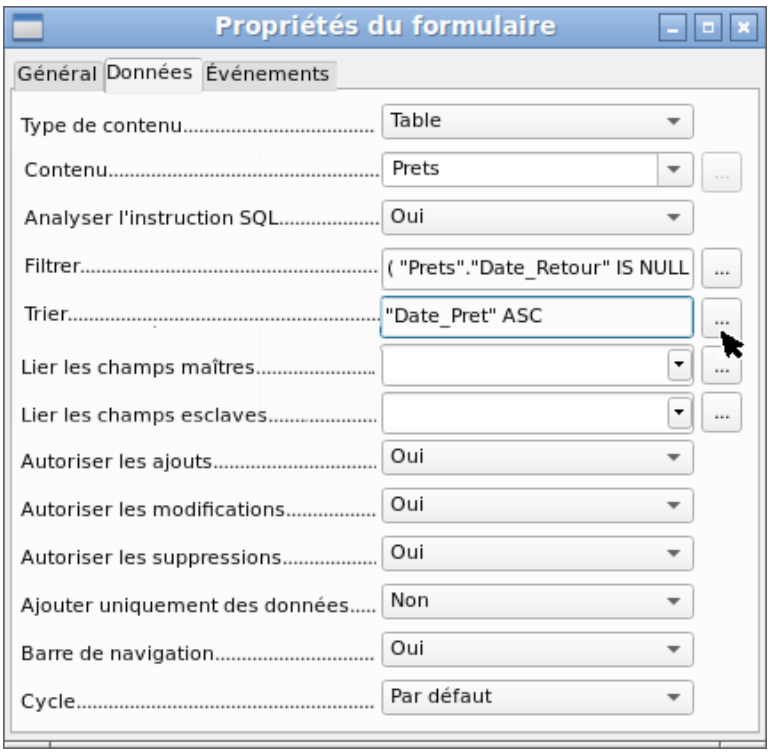


Cliquez avec le bouton droit sur le formulaire principal dans le Navigateur de formulaires pour utiliser le menu contextuel pour créer un nouveau formulaire. Encore une fois, ce formulaire a le nom par défaut de Formulaire, mais il est maintenant un élément dans le sous-dossier du formulaire principal.



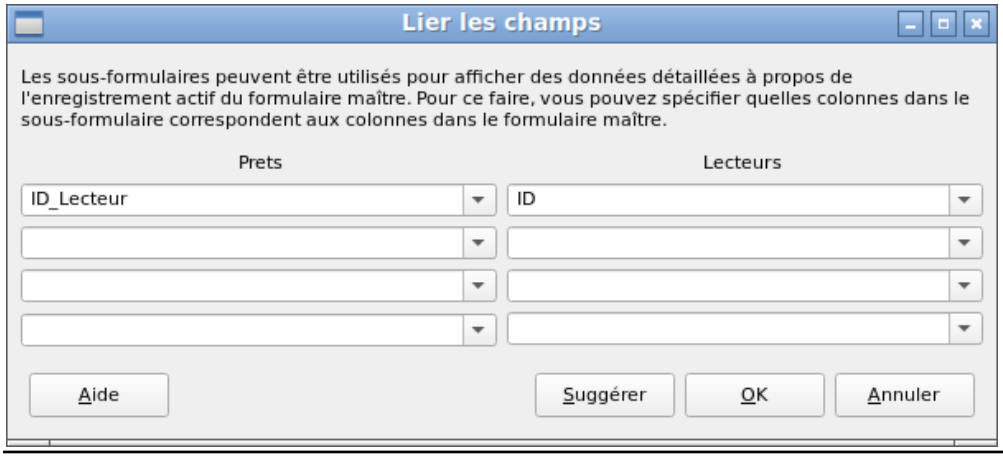


Les propriétés du sous-formulaire doivent maintenant être configurées pour lui donner la bonne source de données, afin de reproduire les données pour le lecteur correct. Il est possible de renommer un formulaire dans l'onglet Général de ses propriétés.



La table Prêts est choisie pour le sous-formulaire. Pour le filtre, nous spécifions que le champ Date_Retour doit être vide ("`Date_Retour" IS NULL`). Cela empêche tout média déjà rendu d'apparaître. Les enregistrements doivent être triés par date de prêt. Le tri croissant montre le support prêté pour la période la plus longue en premier.

Les champs Champs maîtres et Champs esclaves sont utilisés pour créer un lien avec le formulaire principal, dans lequel se trouve le sous-formulaire. Le bouton à trois points montre encore une fois qu'une boîte de dialogue utile est disponible pour les créer.



Sous Prêts, les champs de la table Prêts sont affichés, sous Lecteurs ceux de la table Lecteurs. Le champ ID_Lecteurs de Prets doit être défini comme équivalent au champ ID de la table Lecteurs.

Bien que ce lien ait déjà été créé dans la base de données en utilisant **Outils > Relations** (voir Chapitre 3, Tables), la fonction qui se trouve derrière le bouton **Suggérer** dans cette boîte de dialogue ne fait pas référence à cela et suggère que la première clé étrangère de la table Prêt, à savoir ID_Lecteurs, doit être lié à l'ID de la table Lecteurs. L'assistant de création de formulaire résout mieux ce problème en lisant la relation à partir des relations de la base de données.

Le lien choisi entre la table du sous-formulaire et la table du formulaire principal est désormais spécifié en termes de champs des tables.

Pour créer un contrôle de table dans le sous-formulaire principal, nous devons maintenant sélectionner le sous-formulaire dans le navigateur de formulaires. Ensuite, si l'Assistant Contrôle de table est activé, il affichera les champs disponibles dans le sous-formulaire.

Une fois les champs de table mis en place, nous devons effectuer les modifications déjà évoquées lors de la création du formulaire plus simple :

- Remplacement du champ numérique ID_Media dans le sous-formulaire par une zone de liste.
- Renommer le champ ID_Media en Media.
- Modification des champs numériques pour un format sans décimales.
- Limiter les valeurs minimales et maximales.
- Renommer d'autres champs, pour économiser de l'espace ou pour ajouter des caractères non ASCII qui ne doivent pas être utilisés dans les noms de champs des tables de base de données.

Les fonctions de tri et de filtrage sont complétées pour le formulaire principal par l'ajout d'une barre de navigation. Les autres champs de la barre de navigation ne sont pas nécessaires, car ils sont pour la plupart disponibles depuis le champ table (affichage des enregistrements, navigation dans les enregistrements) ou bien exécutés par déplacement à travers le champ table (stockage des données). La forme finale pourrait ressembler à la figure 21.

Comme précédemment, les champs ID_Genre et ID_Media ont été remplacés par des zones de liste, le type de valeurs par une requête SQL (pour le genre, `SELECT "Genre","ID" FROM "Genres"`) et les étiquettes remplacées.

Voir l'exemple *Form04-VoirPrêtsSelectionTable du fichier Media_sans_Macros*

Figure 21: Formulaire composé d'un formulaire principal (en haut) et d'un sous-formulaire (en bas).

Si un lecteur est maintenant sélectionné dans le formulaire principal, le sous-formulaire affichera les médias prêtés à ce lecteur. Lorsqu'un élément est rendu, il continue à apparaître sur le formulaire jusqu'à ce que le formulaire lui-même soit actualisé. Cela se produit automatiquement lorsqu'un autre enregistrement est chargé dans le formulaire principal. Si le lecteur d'origine est de nouveau sélectionné, les supports rendus ne sont plus affichés.

Cette mise à jour différée est en fait souhaitable dans ce cas, car elle permet d'inspecter les supports actuellement placés sur le compteur de la bibliothèque et de voir immédiatement s'ils ont été enregistrés.

Cette structure de formulaire est beaucoup plus facile à utiliser que la précédente avec un seul formulaire. Cependant, il y a encore des détails qui peuvent être améliorés :

- Les dates de prêt et de média peuvent être modifiées lorsque le média doit être prêté plus longtemps. La modification de la date du support peut rendre impossible le suivi de l'élément encore disponible dans la bibliothèque et de celui qui est prêté. La modification de la date du prêt peut entraîner des erreurs. Les avis de rappel n'ont pas pu être vérifiés.
- Si un enregistrement de lecteur n'est pas sélectionné en cliquant sur l'en-tête d'enregistrement à gauche, seule la petite flèche verte sur l'en-tête montre quel enregistrement est actuellement actif. Il est tout à fait possible que l'enregistrement actif défile directement hors de la fenêtre de contrôle de table.
- Au lieu du texte "Médias prêtés au lecteur sélectionné", il serait préférable d'indiquer le nom du lecteur.
- Il est possible de prêter deux fois le même support sans qu'il ne soit restitué.
- Il est possible de supprimer assez facilement l'enregistrement d'un article prêté.
- Les données peuvent être modifiées ou supprimées dans le formulaire principal. Cela peut être utile pour les petites bibliothèques avec peu de trafic public.
  Cependant, lorsque les choses deviennent mouvementées au comptoir des prêts, l'édition des données des utilisateurs ne doit pas avoir lieu en même temps que l'octroi de prêts. Il

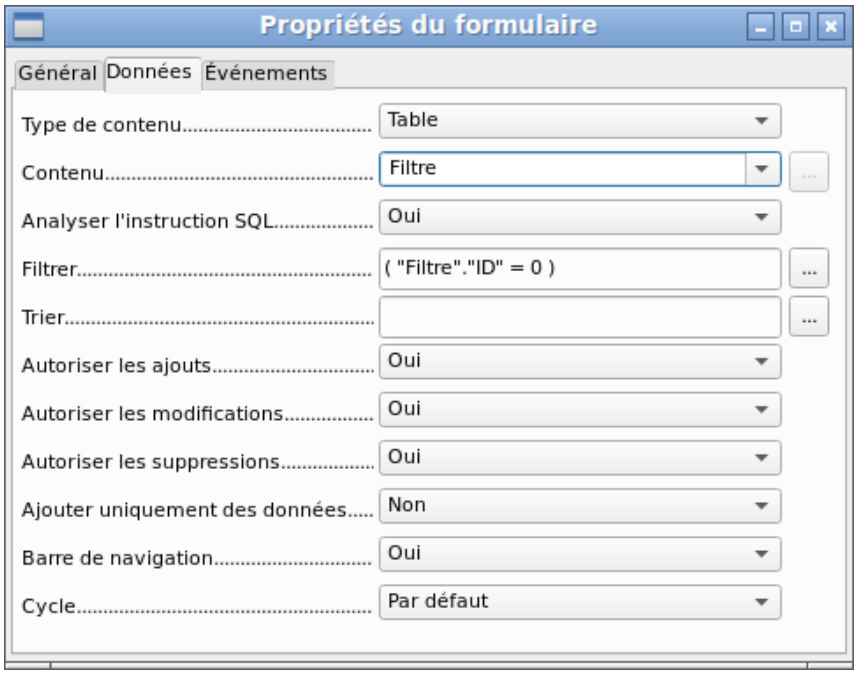
serait préférable que les nouveaux utilisateurs puissent être enregistrés mais que les données utilisateur existantes ne soient pas modifiées. Pour les bibliothèques, cela s'applique également aux suppressions ou aux changements de nom complets.

Améliorons d'abord la sélection des lecteurs. Cela devrait nous protéger des modifications apportées aux registres de prêt. Une solution simple serait de ne permettre aucune modification sauf l'entrée de nouveaux enregistrements. Cela nécessite toujours une fonction de recherche lorsqu'un lecteur souhaite emprunter un article. Il serait préférable d'utiliser une zone de liste pour trouver le lecteur et effectuer les opérations d'émission et de retour dans des champs de table séparés.

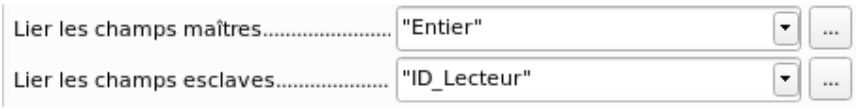
Pour le formulaire principal, nous avons besoin d'une table, dans laquelle la zone de liste peut écrire une valeur liée à cette table. La table nécessite un champ entier et une clé primaire. Il ne contiendra toujours qu'un seul enregistrement, de sorte que l'ID de champ de clé primaire peut être déclaré en toute sécurité comme Tiny Integer. La table suivante nommé Filtre doit donc être créée.

Nom de table : Filtre	
Nom de Champ	***Type de Champ***
ID	Tiny Integer, Primary key
Entier	Integer

La table reçoit une clé primaire avec la valeur 0. Cet enregistrement sera lu et réécrit à plusieurs reprises par le formulaire principal.

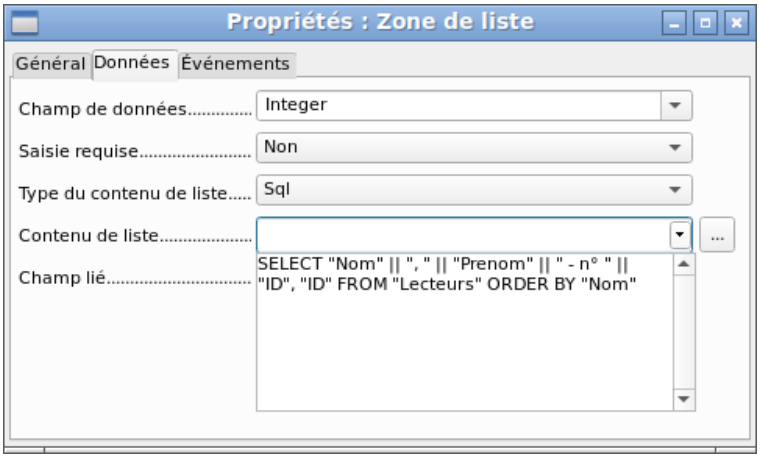


Le formulaire principal est basé sur la table Filtre. Il lira simplement la valeur de la table qui est associée à la clé primaire (ID) de 0. Aucune donnée ne sera ajoutée, l'enregistrement actuel sera simplement réécrit à plusieurs reprises. Comme seules les modifications d'un seul enregistrement sont autorisées, une barre de navigation serait superflue.



Ce formulaire principal est lié au sous-formulaire de telle sorte que la valeur du champ Entier dans la table Filtre est la même que la valeur du champ ID_Lecteur dans la table Prets. Les propriétés du sous-formulaire sont inchangées par rapport à la version ci-dessus.

Avant de créer une zone de liste dans le formulaire principal, nous devons désactiver les assistants. L'assistant de zone de liste vous permet uniquement de créer une zone qui affiche le contenu d'un seul champ. Il serait impossible d'avoir le nom, le prénom et un numéro supplémentaire dans la zone d'affichage d'une zone de liste. Comme dans le formulaire plus simple, nous entrons maintenant pour le contenu de la zone de liste Nom, Prenom – ID Numéro. La zone de liste transmet l'ID à la table sous-jacente.



À côté de la zone de liste, un bouton est créé. Ce bouton fait en fait partie du sous-formulaire. Il reprend deux fonctions : sauvegarder l'enregistrement dans le formulaire principal et mettre à jour la table dans le sous-formulaire. Il suffit de confier la mise à jour au bouton du sous-formulaire. Le processus de sauvegarde du formulaire principal modifié est alors exécuté automatiquement.

Le bouton peut simplement être étiqueté OK. L'action qui lui est assignée dans les propriétés générales du bouton est *Rafraîchir le formulaire*.

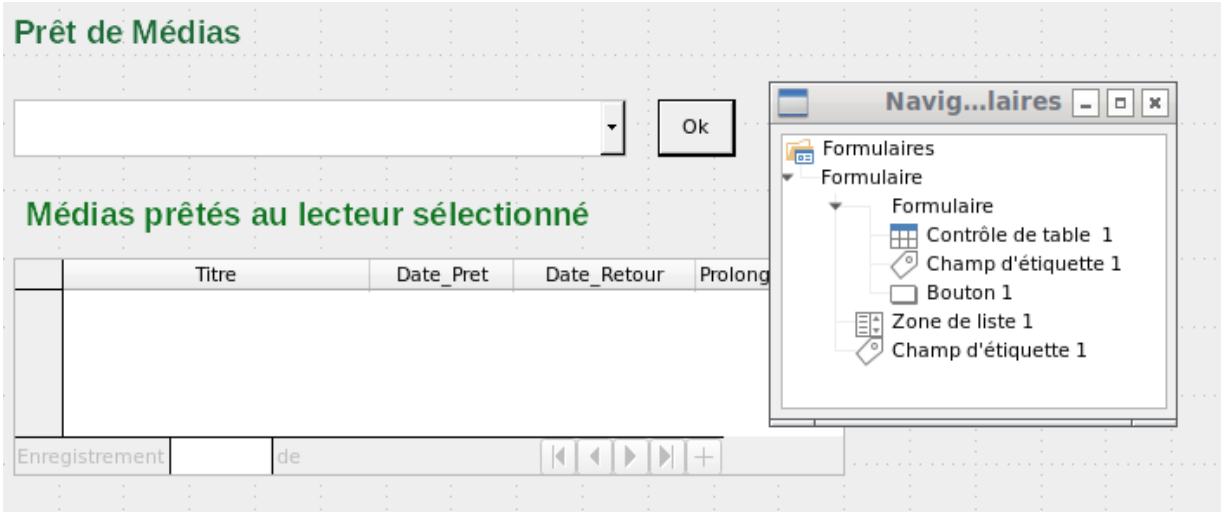


Figure 22: Formulaire principal comme filtre pour un sous-formulaire

Le formulaire principal se compose uniquement de l'en-tête et de la zone de liste ; le sous-formulaire contient un autre en-tête, le champ table de la version précédente et le bouton.

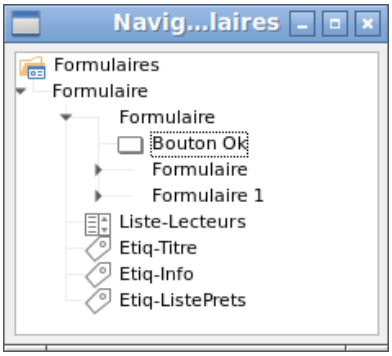
Ce formulaire fonctionne désormais mieux en ce sens que :

- Aucun lecteur ne peut maintenant être édité, modifié ou supprimé, et

- Les lecteurs peuvent être trouvés plus rapidement en tapant une initiale dans le contrôle qu'en utilisant un filtre.

Cet exemple est disponible sous le nom *Form05-VoirPrêtsSelectionListe.*

Pour un plus grand degré de fonctionnalité (retours sans altération des données précédentes) un second sous-formulaire doit être créé, lié à la même table Prêts. Pour garantir la fonctionnalité de la zone de liste de la figure 22, les deux sous-formulaires doivent être placés un niveau plus bas, en tant que sous-formulaires d'un sous-formulaire. Les données sont mises à jour de manière hiérarchique depuis le formulaire principal jusqu'aux sous-formulaires. Le bouton dans le formulaire décrit précédemment doit maintenant être placé dans le premier sous-formulaire et non dans les deux sous-formulaires qui se trouvent en dessous.

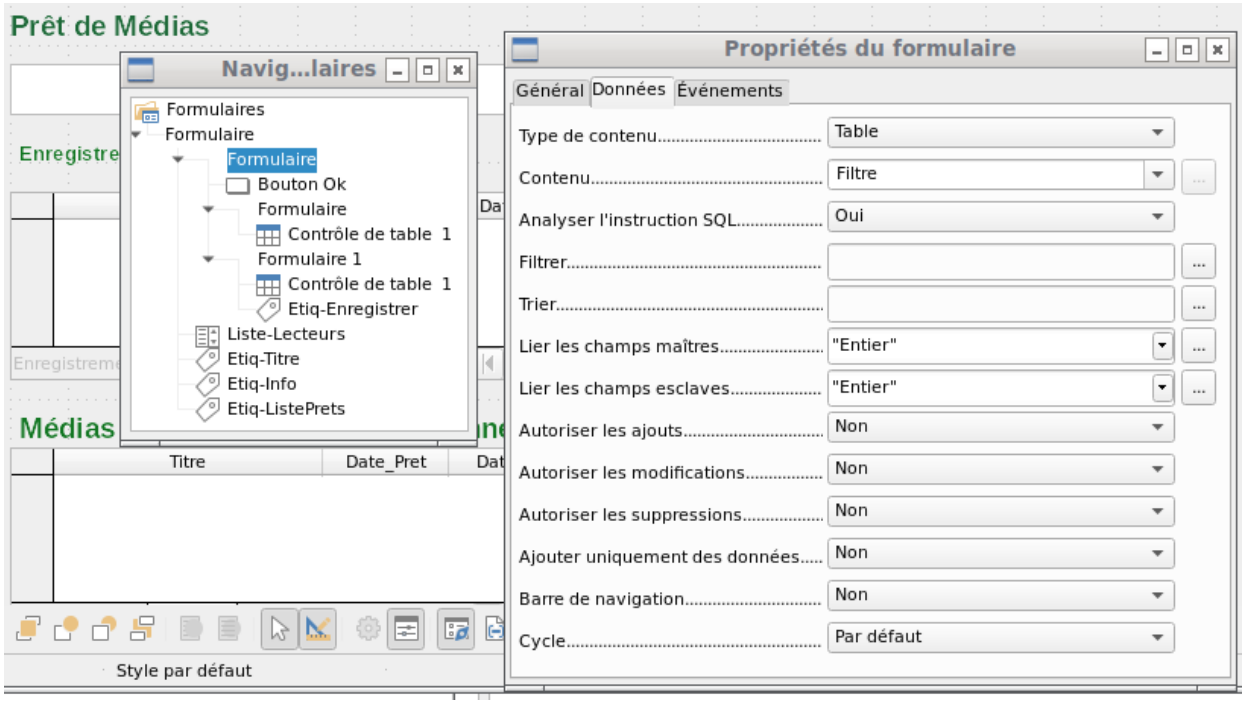


Ici, le navigateur de formulaires est utilisé pour afficher les différents niveaux. Dans le formulaire principal, nous avons le champ d'étiquette pour le titre du formulaire et la zone de liste pour trouver le lecteur. La zone de liste apparaît en bas du formulaire, telle qu'elle est déclarée après le sous-formulaire. Malheureusement, cette séquence d'affichage ne peut pas être modifiée. Le sous-formulaire n'a qu'un seul bouton, pour mettre à jour son contenu et en même temps enregistrer le formulaire principal. Un niveau plus bas se trouvent les deux sous-formulaires supplémentaires. Ceux-ci reçoivent des noms différents lors de leur création afin qu'aucune confusion ne survienne à aucun niveau de l'arbre.

Note

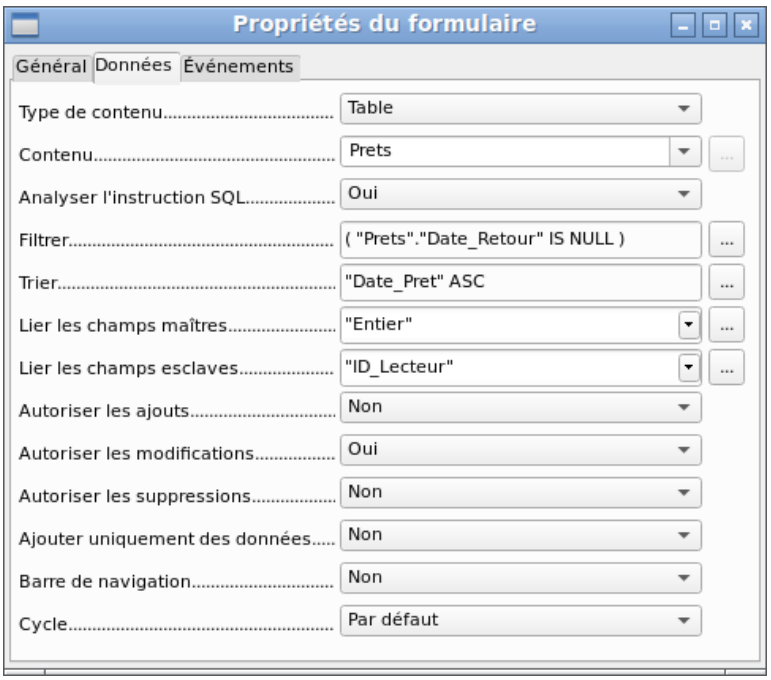
Fondamentalement, les noms des formulaires et des contrôles sont sans importance. Cependant, si ces noms doivent être accessibles par des macros, ils doivent pouvoir être distingués. Vous ne pouvez pas distinguer des noms identiques au même niveau.

Naturellement, il est logique, lors de la création de structures de formulaire plus grandes, d'avoir des noms significatifs pour les formulaires et leurs contrôles. Sinon, trouver le bon champ pourrait rapidement devenir problématique.



Le formulaire principal et le sous-formulaire utilisent la même table. Dans le sous-formulaire, aucune donnée n'est saisie. C'est pourquoi toutes les options de ce formulaire sont définies sur Non. Le formulaire principal et le sous-formulaire sont liés via le champ dont la valeur doit être transmise aux sous-sous-formulaires : le champ Entier dans la table Filtre.

Dans le contrôle table du premier sous-sous-formulaire, aucune donnée existante n'est affichée, il n'est utilisé que pour créer de nouvelles données. Pour cela, le filtre suggéré est adéquat. Seuls les enregistrements correspondant au ID_Lecteur et avec un champ de date de prêt vide ("Date_Pret" IS NULL) seront affichés, donc aucun. En pratique, cela signifie un champ de table vide. Comme le contrôle de table n'est pas mis à jour en permanence, le média nouvellement prêté y restera jusqu'à ce que le bouton de mise à jour OK soit utilisé pour sélectionner un nouveau nom ou pour transférer les données dans le deuxième sous-sous-formulaire.



Le deuxième sous-sous-formulaire nécessite plus de paramètres. Ce formulaire contient également des données de la table “Prets”. Ici, les données sont filtrées pour une date de retour vide. ("Date_Retour" IS NULL). Les données sont triées comme dans le formulaire précédent, de sorte que les supports prêtés depuis plus longtemps soient immédiatement visibles.

Les points suivants sont également importants. Les anciens enregistrements peuvent être modifiés, mais aucun nouvel enregistrement ne peut être ajouté. La suppression est également impossible. Il s’agit de la première étape nécessaire pour éviter que les dossiers de prêt ne soient simplement supprimés ultérieurement. Mais il serait toujours possible de changer le support et la date du prêt. Par conséquent, les propriétés des colonnes devront être ajustées. Finalement, le média et la date du prêt devraient être affichés mais protégés de toute modification.

Le contrôle table est simplement dupliqué après la création du formulaire. Cela se fait en le sélectionnant, en le copiant, en le désélectionnant, puis en le collant à partir du presse-papiers. La copie sera à la même position que l’original et devra donc être déplacée. Après cela, les deux contrôles de table peuvent être modifiés séparément. Le contrôle de table pour le retour de média peut rester pratiquement le même. Seul l’accès en écriture pour les colonnes de média et de date du prêt doit être modifié.

Alors que pour Date_Pret, il est seulement nécessaire de choisir Lecture seule, ce n’est pas suffisant pour les zones de liste. Ce paramètre n’empêche pas la zone de liste d’être utilisée pour apporter des modifications. Cependant, si Activé est défini sur Non, aucun choix ne peut y être effectué. Une zone de liste contenue dans le contrôle de table est ensuite affichée sous la forme d’un champ de texte en lecture seule.

Dans le contrôle de table ci-dessus, tous les champs qui n’ont rien à voir avec le prêt sont supprimés. Seuls le support comme champ de sélection et la date du prêt Date_Pret restent.

Si finalement la requête pour la zone de liste dans le champ de table supérieur est sélectionnée, seuls les médias qui peuvent être prêtés seront affichés. Pour en savoir plus, consultez le chapitre 5, Requêtes.

Voir l’exemple : *Form06-VoirPrêtsEnregistrer du fichier Media_Sans Macros.odb*

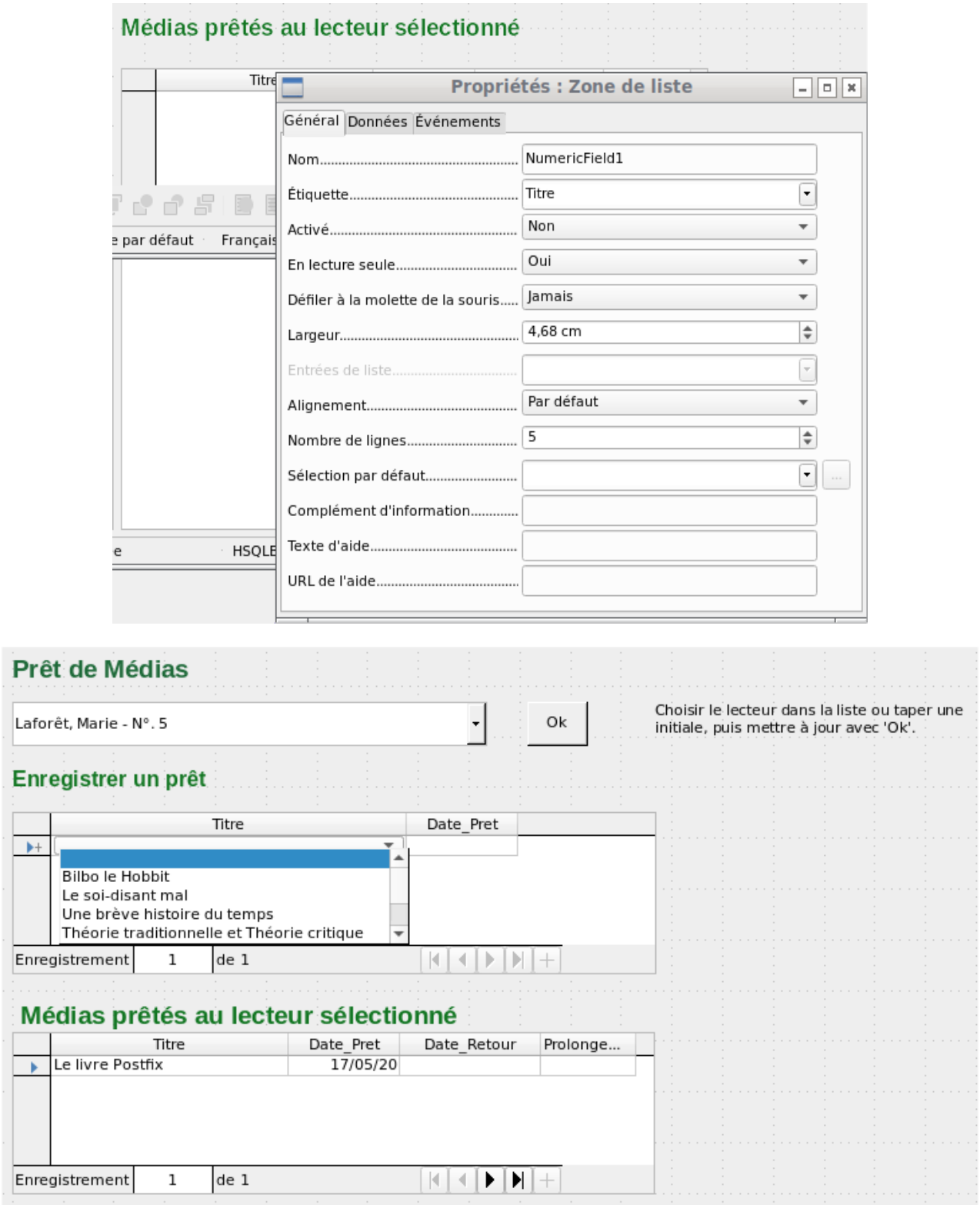
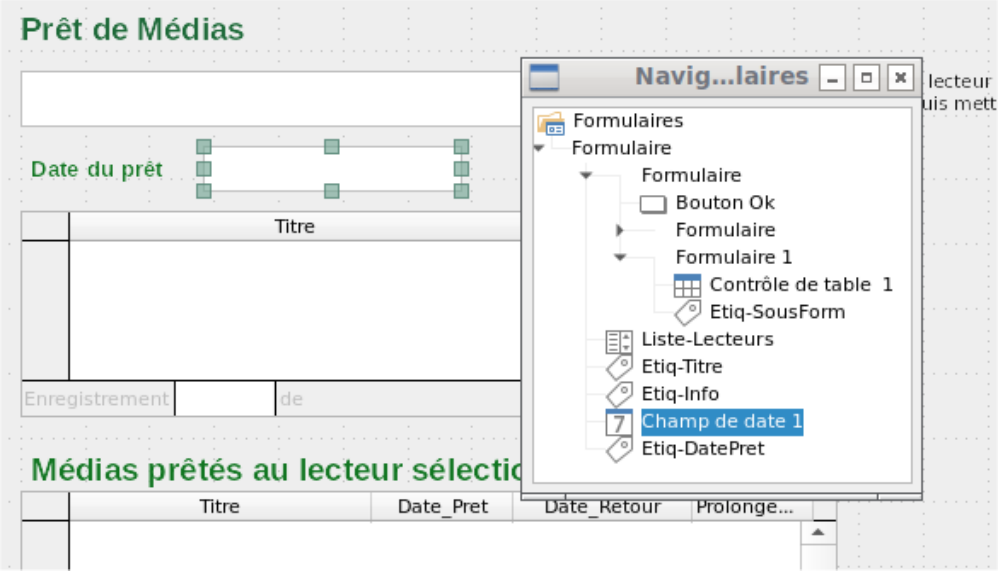


Figure 23: Le champ de sélection dans le sous-formulaire supérieur affiche les supports susceptibles d'être prêtés et permet l'enregistrement.

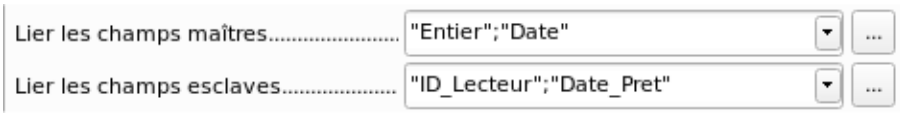
Le formulaire de prêt média est déjà nettement plus utile. Lorsqu'un lecteur arrive au comptoir de prêt, son nom est recherché. Le support à prêter peut-être sélectionné à l'aide de la zone de liste et de la date de prêt saisie. La touche Tab vous amène ensuite à l'enregistrement suivant.

Une dernière amélioration est également souhaitable : actuellement, la date du prêt doit être choisie à chaque fois. Imaginez une journée à la bibliothèque avec peut-être 200 opérations de prêt, peut-être avec une seule personne, qui doit prêter environ 10 médias à chaque fois. Cela nécessiterait la même entrée pour le champ de date encore et encore. Il doit y avoir un moyen de simplifier cela.

Notre formulaire principal est lié à la table Filtre. Le formulaire principal fonctionne uniquement avec l'enregistrement qui a comme clé primaire le "ID" 0. Mais des champs supplémentaires peuvent être créés dans la table Filtre. À l'heure actuelle, aucun champ ne peut stocker une date, mais nous pouvons facilement créer un nouveau champ avec le nom de champ Date et le type de champ Date. Dans la table Filtre, nous avons maintenant stocké non seulement le ID_Lecteur ("Filtre"."Entier") mais également le Date_Pret ("Filtre"."Date").



Dans le formulaire principal, un champ de date supplémentaire apparaît, ainsi qu'une étiquette faisant référence à son contenu. La valeur du champ de date est stockée dans la table Filtre et transférée par les liens du sous-formulaire au sous-sous-formulaire.



Le lien entre les deux formulaires renvoie désormais à deux champs. Le champ Entier est lié au champ ID_Lecteur du sous-sous-formulaire. Le champ Date est lié au champ Date_Pret. Cela garantit que Date_Pret est automatiquement transféré de la table Filtre vers la table Prets lorsque le prêt est effectué.

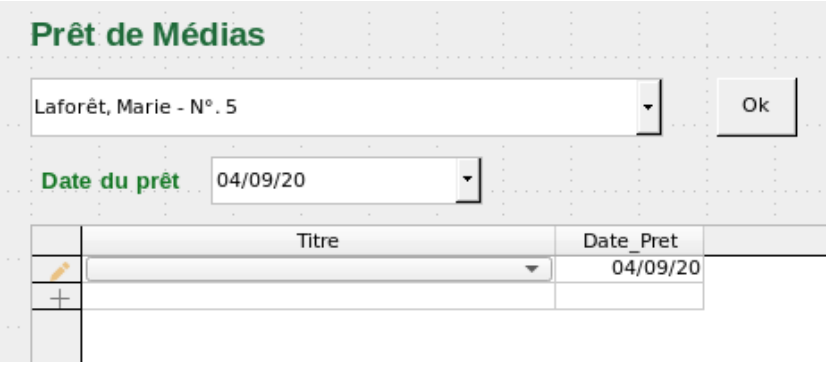
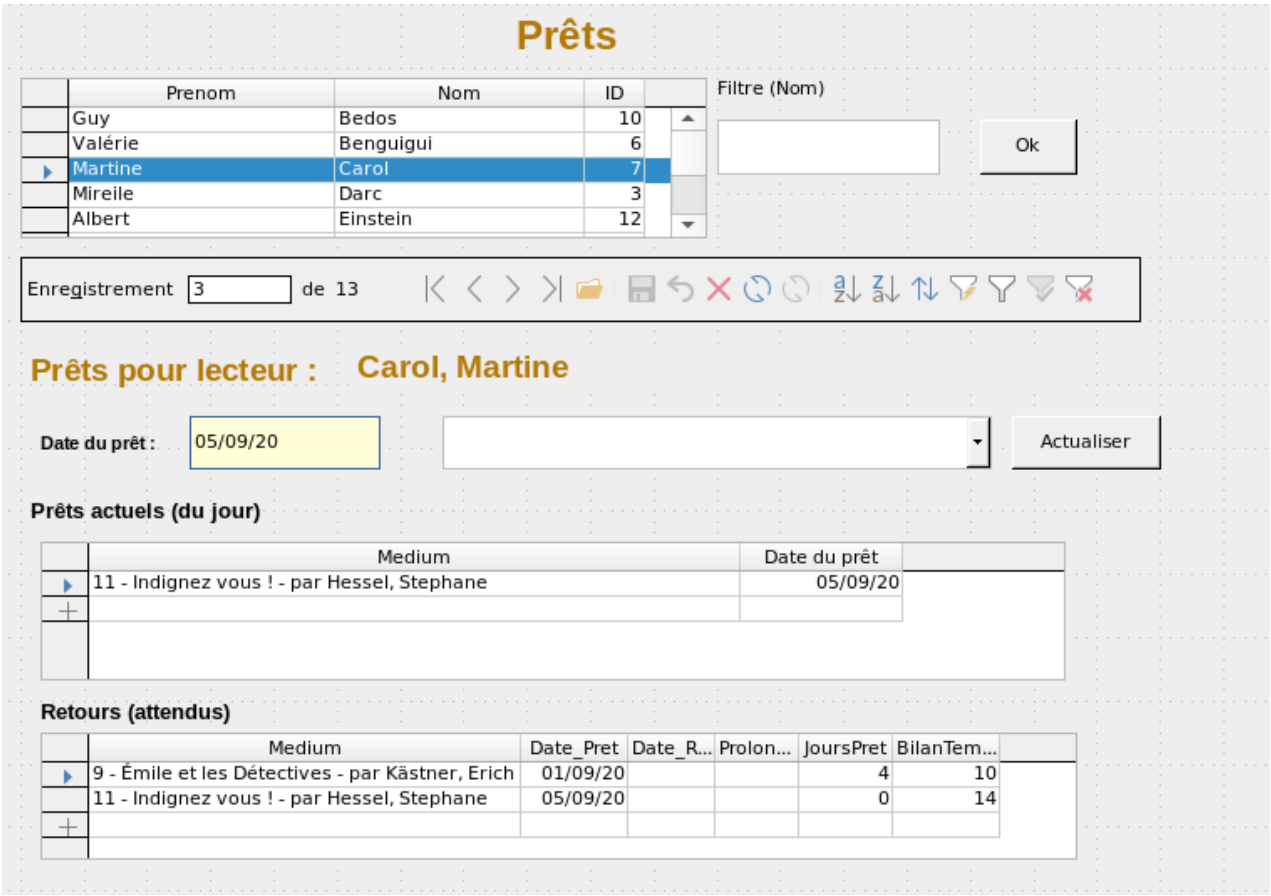


Figure 24: La date du prêt n'est saisie qu'une seule fois. Lorsque le lecteur change, elle doit être ressaisie.

Exemple : *Form07-VoirPrêtsEnregitrerDate*

Le champ de date peut maintenant être supprimé du champ de table, de sorte que ce dernier ne contienne qu'un seul champ de recherche. Ce serait la condition idéale pour accélérer encore plus le travail de la bibliothèque. Car en fait, chaque support aura un numéro d'identification imprimé, alors pourquoi faut-il chercher cela ? Vous pouvez simplement entrer le numéro directement. Ou, mieux encore, il pourrait être numérisé avec un lecteur de code-barres. Ensuite, les médias peuvent être prêtés aussi rapidement que l'emprunteur peut les mettre dans son sac.

Ceci est illustré dans le dernier exemple de formulaire dans la base de données. L'exemple ci-dessus devrait suffire pour comprendre la conception initiale du formulaire mais, comme ce dernier exemple de la base de données, **Media_sans_Macros.odb**, développe le formulaire davantage, les améliorations supplémentaires sont brièvement présentées ci-dessous.



Le formulaire de prêt présente les propriétés suivantes :

- Les lecteurs sont affichés dans un champ de table. Ici, vous pouvez également entrer de nouveaux lecteurs.
- À l'aide d'un filtre, lié à la table Filtre, les noms peuvent être filtrés à l'aide de leur lettre initiale. Ainsi, si vous saisissez A, seules les personnes dont le nom commence par A seront affichées. Ce filtrage est indépendant de la casse.
- Le sous-titre montre à nouveau le nom de la personne à qui le prêt doit être fait. Si un verrou a été placé sur cet emprunteur, il est également affiché.
- La date du prêt est définie sur la date du jour. Cela se fait dans la table de filtrage en utilisant SQL de telle sorte que, lorsqu'aucune date n'est entrée, la valeur par défaut à stocker est la date actuelle.
- Les supports empruntables sont sélectionnés à l'aide d'une zone de liste. Lorsque vous appuyez sur le bouton Mettre à jour, le prêt est transféré vers le contrôle de table au-dessous.
- Le champ de table au milieu sert uniquement à afficher la date réelle du prêt pour le média. Ici, il est également possible de corriger une erreur rétrospectivement en supprimant la ligne.
- Dans le contrôle de table inférieur, comme dans l'exemple ci-dessus, la modification des supports et des dates de prêt n'est pas possible. Il n'est pas non plus possible de supprimer des enregistrements.
- Outre la saisie de la date de retour ou, le cas échéant, une prolongation du prêt, ce tableau affiche également le nombre de jours pour lesquels le support peut être prêté et le nombre de jours restants de la période de prêt en cours.
- Si ce temps restant devient négatif, le support doit être renvoyé immédiatement. Le compte est alors verrouillé. Cela ne redevient possible que lorsque le support est renvoyé. Après le retour, le bouton Mettre à jour ne doit être appuyé qu'une seule fois.

Ce formulaire, créé à l'aide de requêtes, est beaucoup plus complexe dans sa structure que la version précédente. Vous pouvez en savoir plus sur l'essentiel dans le chapitre 5, Requêtes.

Une vue – plusieurs formulaires

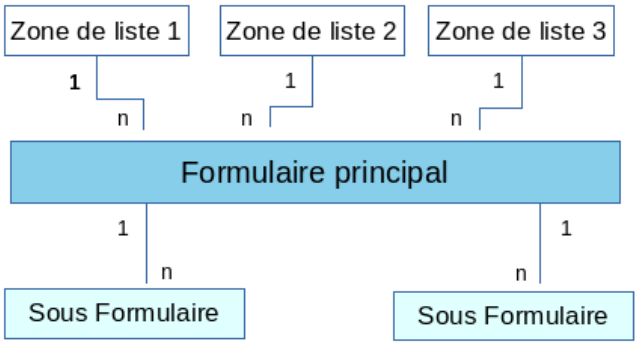
Alors que l'exemple du formulaire Prêts n'implique que des entrées dans une seule table (la table Prets) et permet en outre l'entrée dans le sous-formulaire pour les nouveaux lecteurs, la procédure d'entrée pour les médias est beaucoup plus étendue. Dans sa forme finale, la table des médias est entourée d'un total de huit tables supplémentaires (voir Chapitre 3, Tables).

Les tables SousTitre et relation_Media_Auteur sont liées aux sous-formulaires du formulaire Media via une relation n:1. En revanche, les tables avec une relation 1:n avec la table Media doivent être représentées par des formulaires situés au-dessus de la table Media. Comme il existe plusieurs de ces tables, leurs valeurs sont saisies dans le formulaire principal à l'aide de zones de liste.

La table d'un formulaire principal et la table d'un sous-formulaire sont dans une relation 1:n, ou dans certains cas exceptionnels 1:1. Par conséquent, après une longue utilisation de la base de données, le formulaire principal gère généralement une table qui contient beaucoup moins d'enregistrements que la table appartenant au sous-formulaire.

Plusieurs formulaires principaux ne peuvent pas inclure le même sous-formulaire. Par conséquent, il n'est pas possible de créer une disposition de formulaire pour de nombreuses relations 1;n simultanées dans lesquelles le sous-formulaire a le même contenu. Lorsqu'il existe une relation 1:n pour la table appartenant à un formulaire, vous pouvez utiliser une zone de liste.

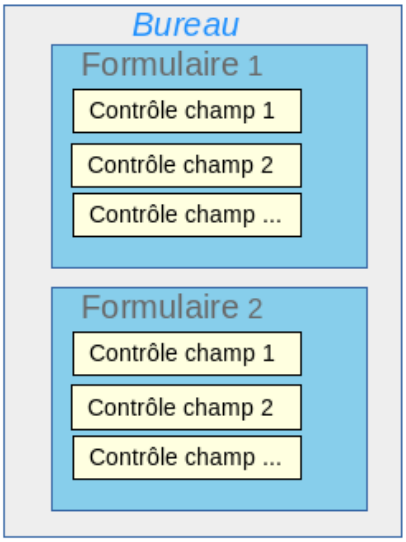
Ici, seuls quelques termes sont disponibles à partir d'une autre table, ceux dont les clés étrangères peuvent être saisies dans la table du formulaire principal de cette manière.



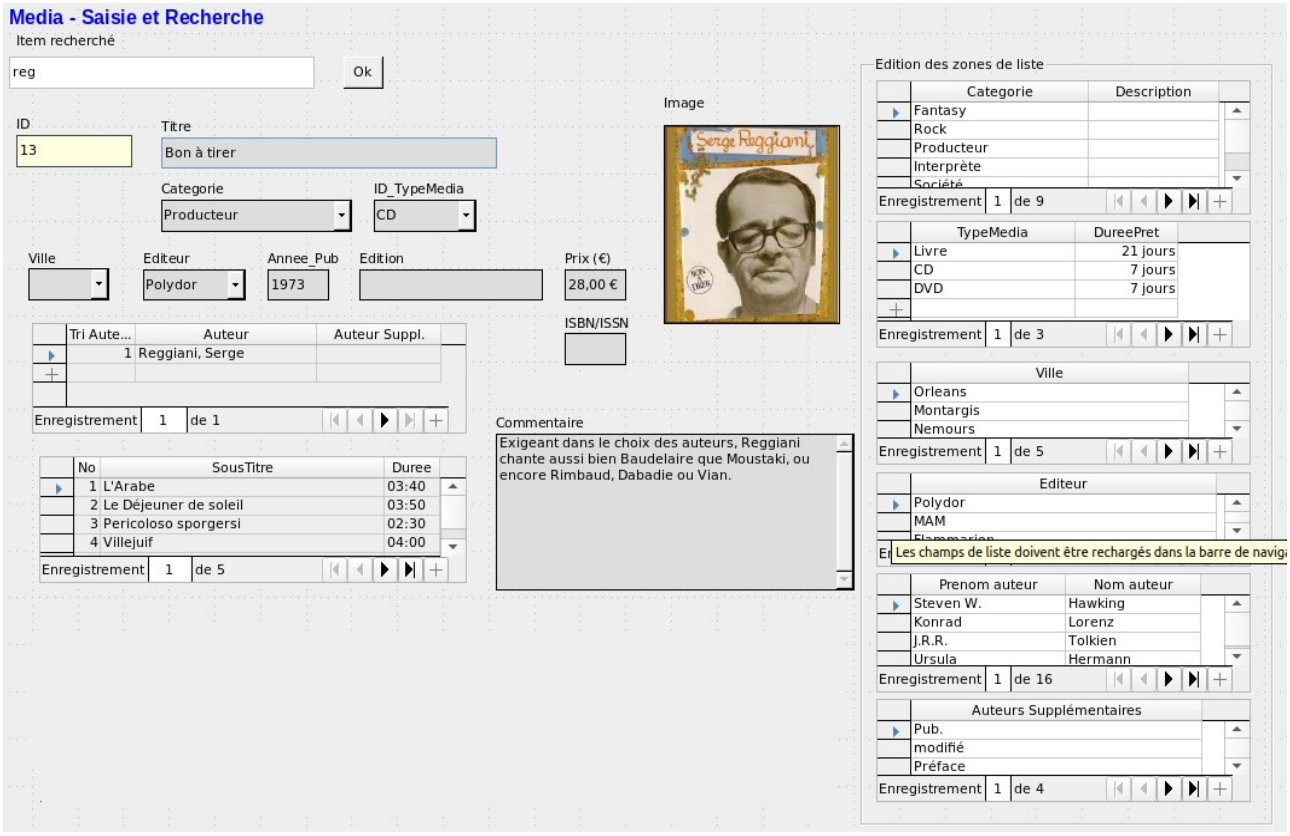
À l'aide de zones de liste, le formulaire principal basé sur la table Media peut recevoir des valeurs des tables Categories, Ville ou Editeurs. Les sous-formulaires sont utilisés pour lier les tables relation_Media_Auteur et SousTitre avec le formulaire principal et à travers celui-ci avec la table Media.

Le sous-formulaire de la table relation_Media_Auteur se compose à nouveau de deux zones de liste afin que les clés étrangères des Auteurs et ID_Auteur_Suppl (les ajouts peuvent être par exemple éditeur, photographe, etc.) ne soient pas entrées directement sous forme de nombres.

Pour le formulaire de saisie des médias, les zones de liste doivent généralement être remplies progressivement au cours du processus de saisie. À cette fin, d'autres formulaires sont intégrés à côté du formulaire principal. Ils existent indépendamment de la forme principale.

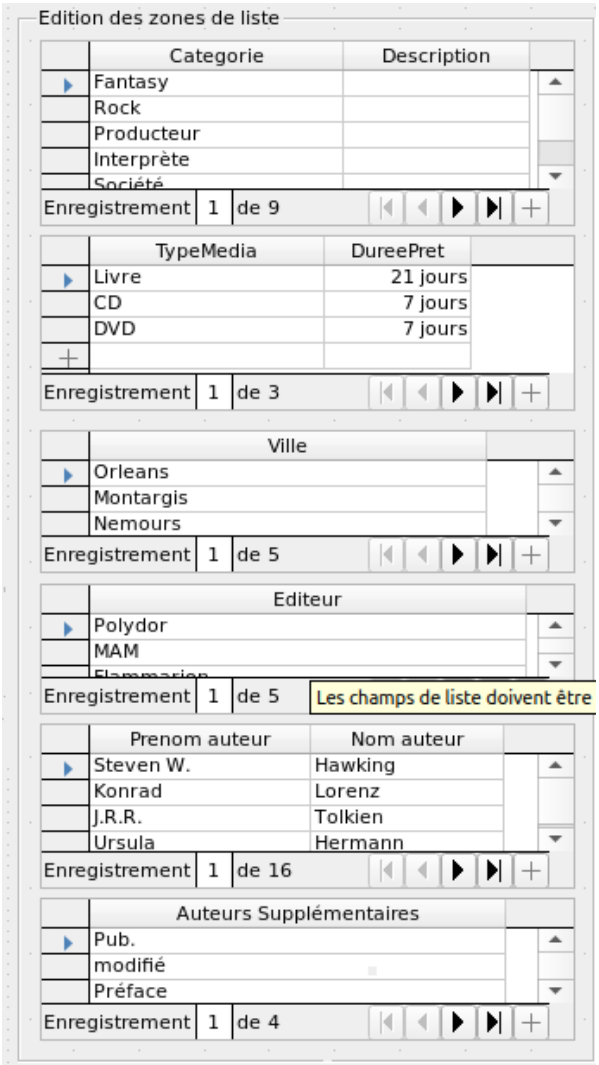


La forme générale de saisie des médias ressemble à ceci :



Sur le côté gauche se trouve le formulaire principal en vue de la recherche et de la saisie de nouveaux médias. Sur le côté droit se trouve une zone de groupe avec l'étiquette Modifier le contenu de la zone de liste, fournissant une zone séparée destinée à remplir les zones de liste dans le formulaire principal. Si la base de données n'existe pas depuis longtemps, il sera souvent nécessaire de faire des entrées dans ces champs. Toutefois, plus il y a d'entrées disponibles pour les zones de liste du formulaire principal, moins il sera nécessaire d'accéder aux contrôles de table dans la zone de groupe.

Les contrôles de table suivants sont tous subordonnés en tant que formulaires secondaires individuels au formulaire principal, le formulaire d'entrée.



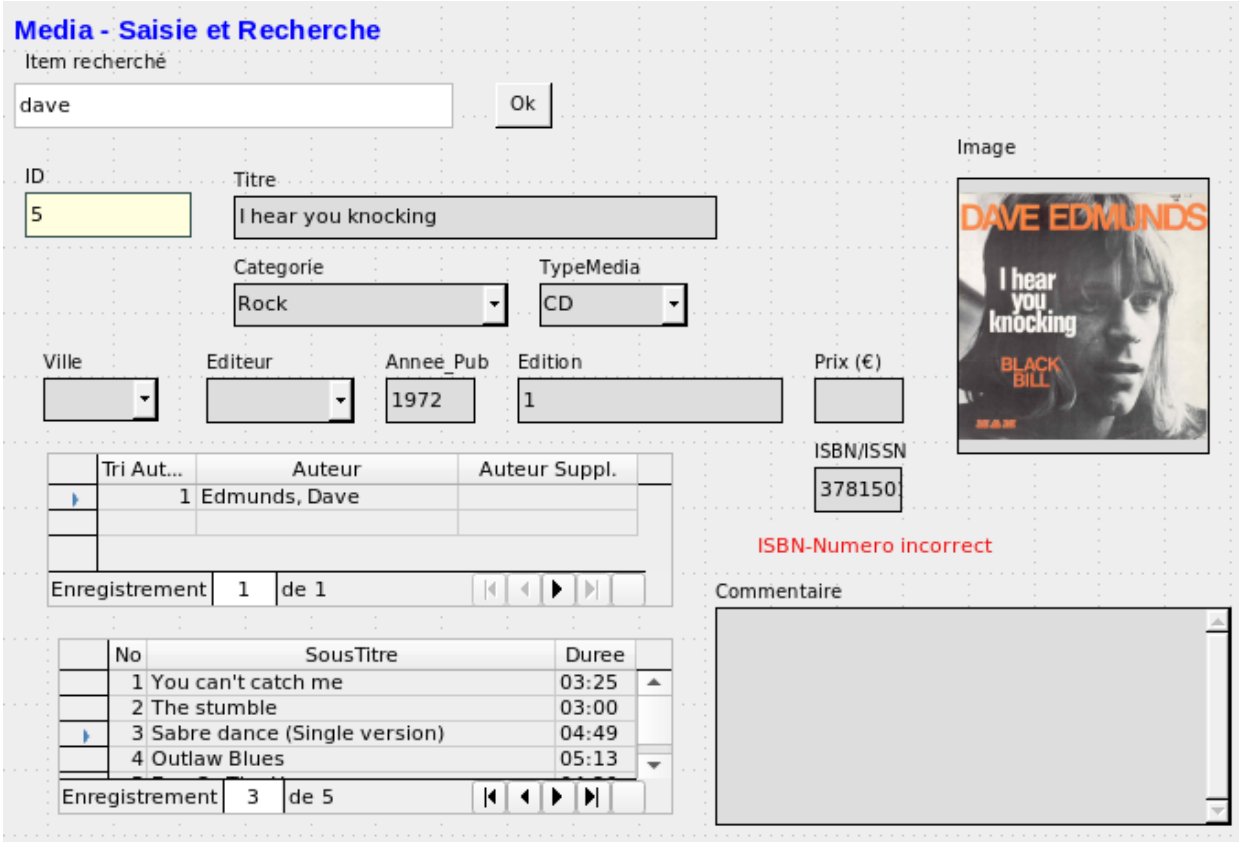
Ici, dans chaque cas, les données complètes d'une table sont saisies. Dans les premiers temps, il est souvent nécessaire d'avoir recours à ces formulaires annexes, car peu d'auteurs sont encore stockés dans la table correspondante.

Lorsqu'un nouvel enregistrement est stocké dans l'un des contrôles de table, il est nécessaire de rechercher la zone de liste correspondante dans le formulaire principal et d'utiliser le contrôle de mise à jour (voir Barre de navigation) pour lire les nouvelles valeurs.

Le Navigateur de formulaires affiche une grande liste correspondante de formulaires.

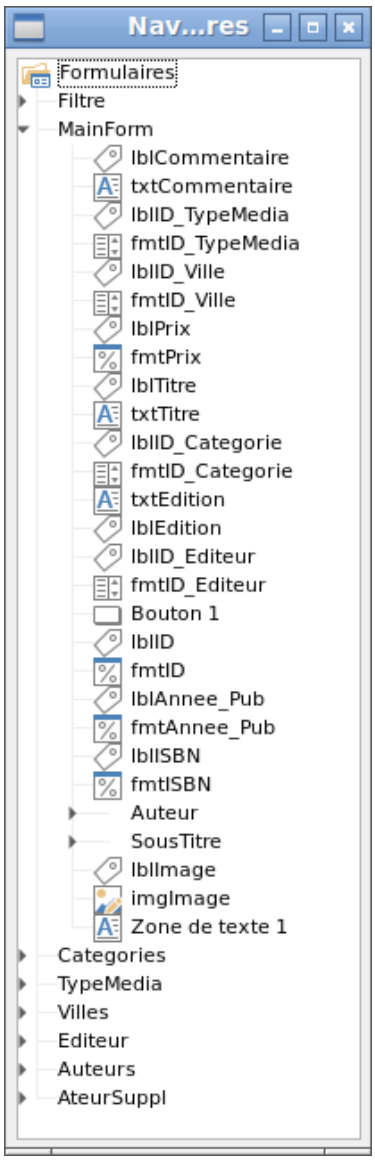
Les formulaires ont été nommés individuellement pour faciliter la reconnaissance. Seul le formulaire principal a toujours le nom de MainForm qui lui a été donné par l'assistant. Au total, il existe huit formulaires parallèles. Le formulaire Filtre héberge une fonction de recherche tandis que le formulaire MainForm est l'interface d'entrée principale. Tous les autres formulaires concernent l'un ou l'autre des champs de table présentés ci-dessus.

Sans les contrôles de table, le formulaire principal semble un peu plus simple.



Le champ du terme recherché se trouve dans le formulaire Filtre, les deux champs de table (pour l'auteur et le sous-titre) se trouvent dans le sous-formulaire du formulaire principal Saisie média.

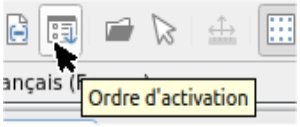
Dans le navigateur de formulaires, ce formulaire semble beaucoup plus complexe, car tous les contrôles et étiquettes y apparaissent. Dans le formulaire précédent, la plupart des champs étaient en fait des colonnes dans les contrôles de table et étaient donc invisibles pour le navigateur de formulaire.



Malheureusement, la séquence dans le navigateur de formulaire ne peut pas être facilement modifiée. Ainsi, par exemple, il semblerait plus judicieux de placer les sous-formulaires SousTitre et Auteur en tant que branches de MainForm dès le début. Mais dans le navigateur de formulaires, les contrôles et sous-formulaires individuels sont simplement affichés dans l'ordre dans lequel ils ont été créés.

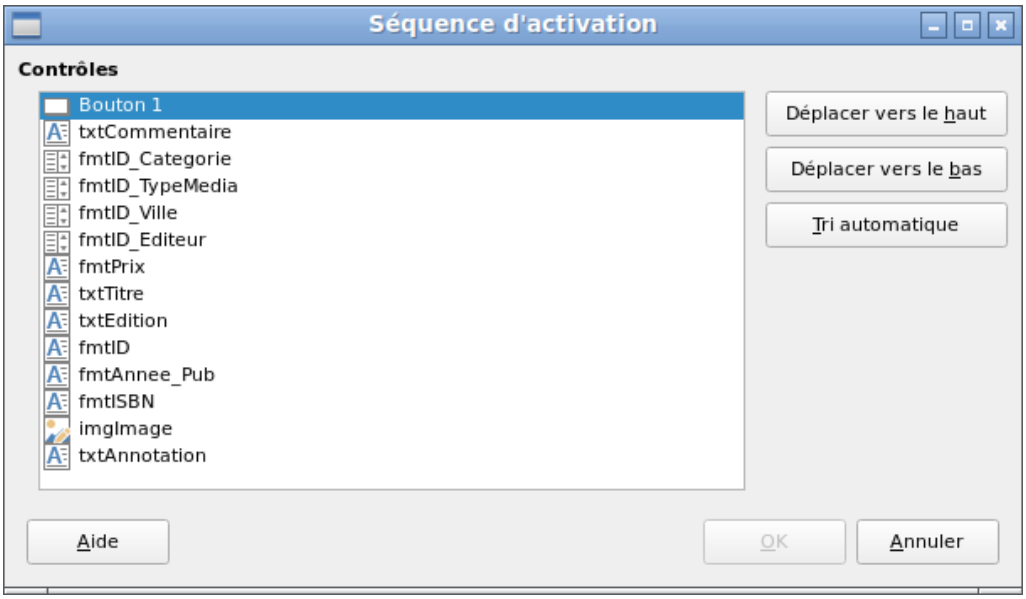
L'Assistant Formulaire fournit aux contrôles des abréviations particulières qui apparaissent à côté des icônes et indiquent de quel type de contrôle il s'agit. Les libellés (étiquettes) commencent par 'lbl', les champs de texte par 'txt' et ainsi de suite. Dans l'ensemble, les étiquettes ne fournissent que des informations secondaires pour la saisie des données. Ils peuvent également être placés directement au-dessus des blocs de texte et n'apparaissent pas dans le navigateur de formulaires.

La séquence des éléments dans le navigateur n'a rien à voir avec la séquence de tabulations. Cela est déterminé par l'ordre d'activation.



La séquence d'activation pour un formulaire particulier est appelée lorsqu'un élément du formulaire est sélectionné et que le bouton Ordre d'activation est cliqué.

Pour un formulaire simple, il n'est pas nécessaire de faire les choses dans cet ordre, mais là où il existe de nombreux formulaires parallèles, la fonction a besoin de savoir quel formulaire est prévu. Sinon, par défaut, il s'agit du premier formulaire du navigateur de formulaires.



La séquence d'activation permet à tous les éléments qui transmettent des données à la table sous-jacente du formulaire ou peuvent effectuer des actions, d'être mis en ordre. Cela correspond au paramétrage de la séquence d'activation dans les propriétés de contrôle listées à la page 14.

Notez que dans la séquence d'activation, certaines commandes apparaissent pour lesquelles le taquet de tabulation est en fait désactivé. Ils sont inclus dans la liste, mais ne sont en fait pas accessibles via le clavier lorsque vous travaillez avec le formulaire.

Les champs peuvent être automatiquement triés dans l'ordre dans lequel ils apparaissent sur l'arrière-plan du formulaire. Plus un champ est haut, plus il arrive tôt dans la séquence lorsque vous utilisez le tri automatique. Pour les champs de même hauteur, le champ le plus à gauche vient en premier. Ce tri fonctionne sans erreur uniquement lorsque les éléments ont été positionnés exactement à l'aide de la grille lors de la création du formulaire. Sinon, vous devrez les ajuster. Sélectionnez simplement un contrôle et utilisez *Déplacer vers le haut* ou *Déplacer vers le bas* pour le déplacer vers le haut ou vers le bas dans la séquence.

S'il existe un sous-formulaire, le tri automatique saute directement dans le sous-formulaire après avoir parcouru le formulaire principal. Dans le cas d'un contrôle de table, pendant la saisie au clavier le curseur est piégé dans ce sous-formulaire ; vous ne pouvez le libérer qu'en utilisant la souris ou en appuyant sur Ctrl + Tab.

Le tri automatique ne fonctionne qu'une seule fois pour un contrôle de table. Un sous-formulaire suivant avec un contrôle de table ne sera pas inclus. Les formulaires parallèles ne sont pas non plus pris en compte. Un tri automatique ne peut pas être effectué rétrospectivement pour un sous-formulaire avec un contrôle de table. Le sous-formulaire doit être complètement supprimé (temporairement déplacé vers un autre formulaire).

Le substrat de données pour le formulaire de saisie multimédia n'est pas dans ce cas une table mais une requête.

Ceci est nécessaire, car le formulaire doit être utilisé non seulement pour saisir des enregistrements, mais également pour la recherche. Le formulaire contient également un champ de texte qui indique, après enregistrement, si le numéro ISBN saisi était correct. Cela aussi n'est possible qu'en utilisant une requête complète. Pour en comprendre le contexte, il est donc nécessaire de discuter des principes fondamentaux des requêtes, traités au chapitre 5.

Messages d'erreur lors de la saisie dans les formulaires

Certains messages d'erreur qui se produisent généralement lors de la conception initiale d'un formulaire sont brièvement expliqués ici.

Attempt to insert null into a non-nullable column : NomColonne : ID table : NomTable in statement…

Certains champs ne peuvent pas être vides. S'ils sont ainsi définis dans la table (*NOT NULL*), vous obtenez ce message d'erreur. S'il a également été déclaré non nul dans le formulaire, vous voyez également un message en langue native avec le nom exact du champ qui doit être rempli.

Le message ci-dessus apparaît très souvent lorsque la clé primaire, dans ce cas ID, n'a pas été importée dans le formulaire. Il a été conçu comme un champ à incrémentation automatique, mais malheureusement le concepteur a oublié de le définir comme Valeur automatique. Vous devez donc corriger cela ou le champ doit apparaître dans le formulaire, afin que vous puissiez saisir une valeur.

Définir un champ rétrospectivement comme Valeur automatique est parfois problématique, si la table a des relations avec d'autres tables ou lorsque la table est accessible à l'aide d'une vue. Tous les liens doivent être supprimés pour rendre le champ de clé primaire de la table modifiable. Le contenu de la commande SQL qui a été utilisée pour créer la vue peut être stocké temporairement dans une requête.

Integrity constraint violation – no parent SYS_FK_95 table : NomTable in statement…

Ici, nous avons un lien entre la table sous-jacente au formulaire et une autre table. Cette table est censée fournir sa clé primaire à utiliser dans un champ de clé étrangère. Normalement, cela se fait en utilisant une zone de liste qui exécute la requête correcte pour récupérer la clé. Si vous n'utilisez pas une zone de liste ou si la zone de liste n'a pas été correctement construite, le champ de clé étrangère peut acquérir une valeur incorrecte qui n'est pas réellement présente dans la table source en tant que clé primaire. C'est ce que l'on entend par "violation de contrainte d'intégrité". '*No parent SYS_FK_95*' signifie que dans la deuxième table, la table source "Parents", la valeur d'index correspondante portant le nom "*SYS_FK_95*" n'est pas présente.

. Errors when entering new records
Errors in running functions

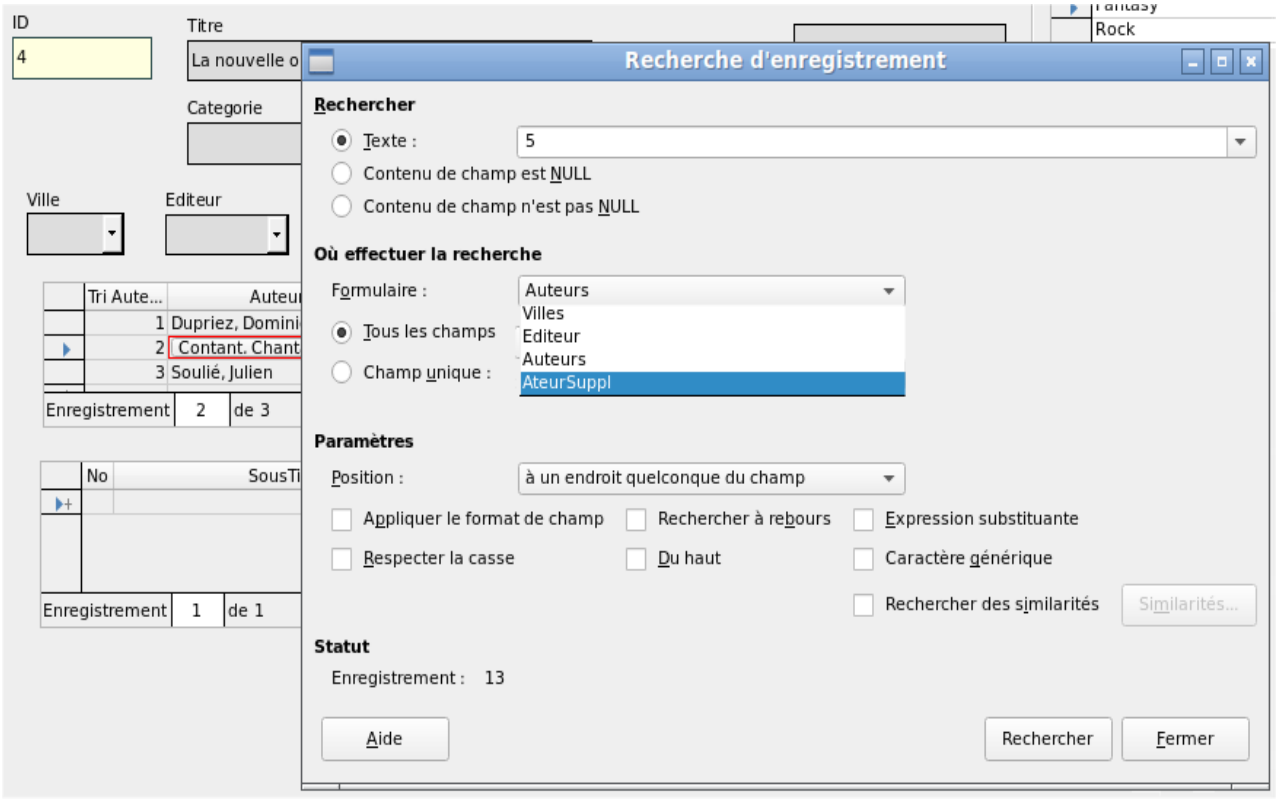
Supposons qu'un formulaire fournisse des données à un sous-formulaire. Base ne voit pas cela comme une modification de la valeur d'un champ. L'interface utilisateur graphique propose de sauvegarder la valeur mais l'enregistrement apparaît vide. La réponse est d'inclure un champ simple, par exemple un champ oui / non, dans la table sous-jacente. Maintenant, avant de sauvegarder, supprimez ce champ et l'enregistrement peut être sauvegardé sans aucun problème.

Les valeurs transmises via d'autres formulaires ne sont apparemment saisies dans les champs correspondants que lorsqu'il y a au moins une action supplémentaire dans le formulaire.

Recherche et filtrage dans les formulaires à l'aide de la barre de navigation

La barre de navigation du formulaire offre diverses possibilités de recherche et de filtrage. Les éléments individuels de la barre de navigation ont déjà été répertoriés ci-dessus.

Recherche d'enregistrement à l'aide de paramètres



Une simple recherche d'enregistrements avec des paramètres est décrite dans le chapitre 3, Tables. Elle offre un nombre considérable de paramètres. Une caractéristique de ce type de recherche est que, fondamentalement, tous les enregistrements sont tamisés plutôt que de limiter les enregistrements affichés à ceux contenant les paramètres de recherche.

Les recherches d'enregistrements dans les formulaires vous permettent de rechercher à la fois dans le formulaire et tous les sous-formulaires, mais il n'est pas possible de rechercher tous les formulaires à la fois.

L'illustration montre l'accès au sous-formulaire Form_Auteur dans le formulaire principal MainForm. Il s'agit d'un champ de liste contenant les noms des auteurs. Cependant, ce qui est recherché n'est pas le texte dans le champ de liste mais les valeurs qui sont entrées comme clés étrangères lorsque le champ est utilisé.

Malheureusement, ce type de recherche présente les inconvénients suivants :

La recherche de paramètres est trop compliquée pour une utilisation normale.

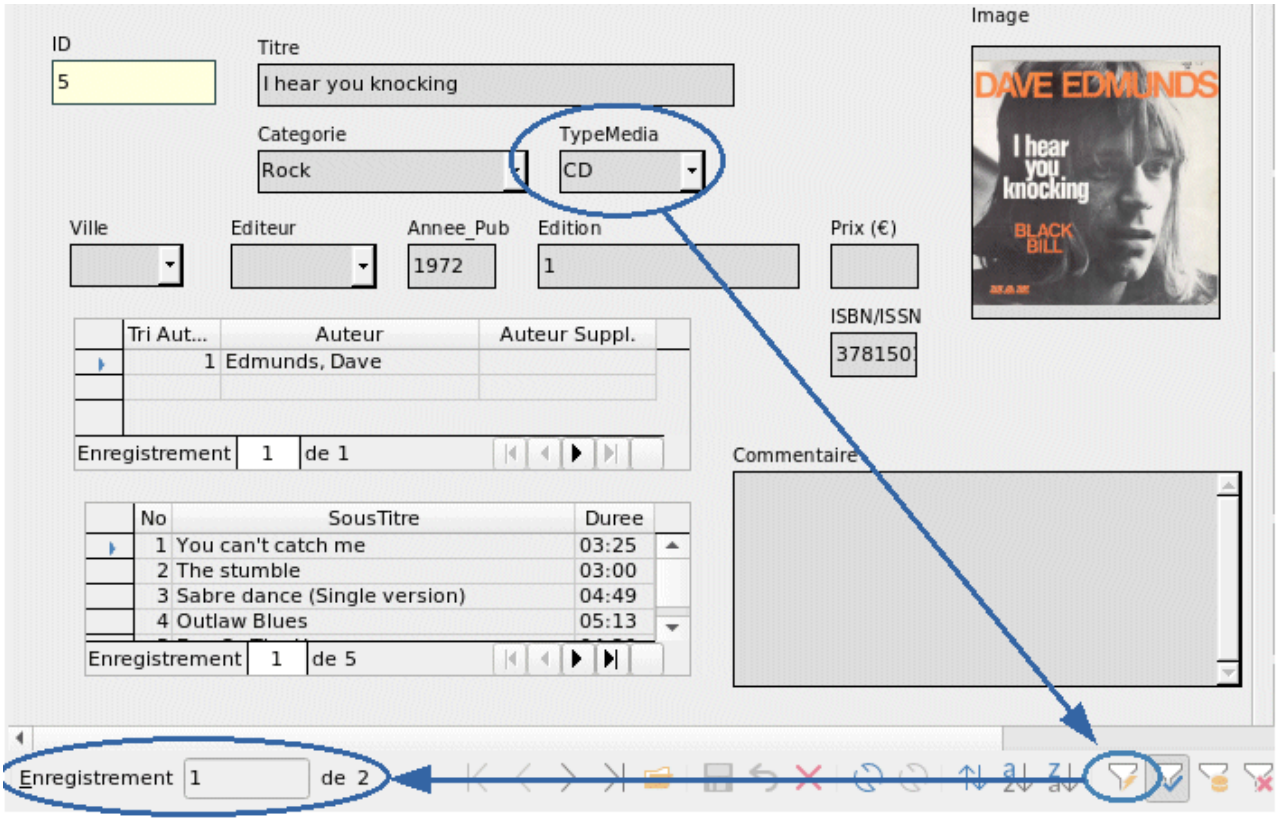
- La fonction de recherche elle-même est très lente, car elle n'utilise pas les fonctions de requête de la base de données.
- La recherche ne fonctionne que pour un formulaire à la fois, mais pour tous les champs. Les sous-formulaires doivent être examinés séparément.
- Nous recherchons dans un sous-formulaire qui appartient au formulaire principal actuel. Les inscriptions dans d'autres sous-formulaires ne seront pas révélées.

Ainsi, par exemple, il n'est pas possible de trouver un sous-titre d'un support autre que celui actuellement affiché.

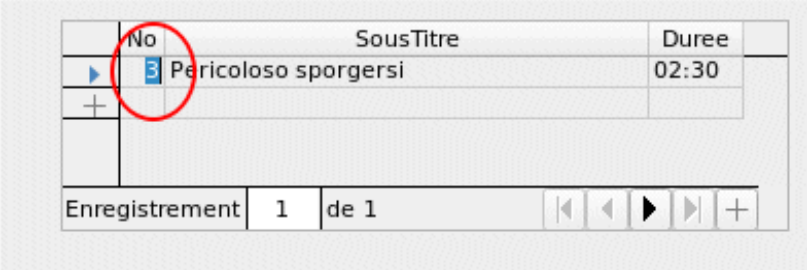
- La recherche s'applique aux fonctions de champ de liste sur la base de champs clés (clés étrangères) stockées dans la table. Les données affichées ne peuvent pas être utilisées.

Filtrer avec l'autofiltre

Le filtre automatique peut être sélectionné directement dans la barre de navigation. Cliquez sur un champ dans le formulaire principal et il filtrera sur le contenu de ce champ. Le filtre automatique fonctionne également avec des champs de liste et présente un avantage évident par rapport à la saisie manuelle de clés étrangères.



Dans l'exemple ci-dessus, un enregistrement contenant "CD" dans le champ TypeMedia est récupéré. Ensuite, le bouton de filtre automatique est basculé. Sur les 9 enregistrements d'origine de la table Media, seuls 2 enregistrements s'affichent désormais dans le compteur d'enregistrements.



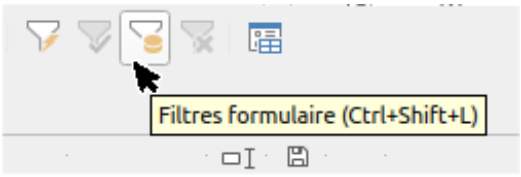
Bien que seuls deux enregistrements aient un "3" dans le champ No°, les principaux formulaires de tous les autres enregistrements continuent d'être affichés. Dans le sous-formulaire, seuls les enregistrements correspondant à "3" sont affichés.

Cependant, l'utilisation du filtre automatique ne peut pas résoudre les problèmes suivants :

- Si vous recherchez dans un sous-formulaire, seule la valeur spécifiée y sera affichée, mais le formulaire principal est affiché pour tous les enregistrements même s'ils ne contiennent pas cette valeur. Ainsi, par exemple, si vous recherchez un support avec un deuxième auteur, vous verrez toujours tous les supports qui n'ont qu'un seul auteur. Dans ces enregistrements, le champ auteur sera vide.
- La valeur utilisée pour le filtre doit être spécifique. Il n'y a aucune possibilité de recherche de similitude pour trouver des matériaux connexes.

Filtrage avec le filtre basé sur le formulaire

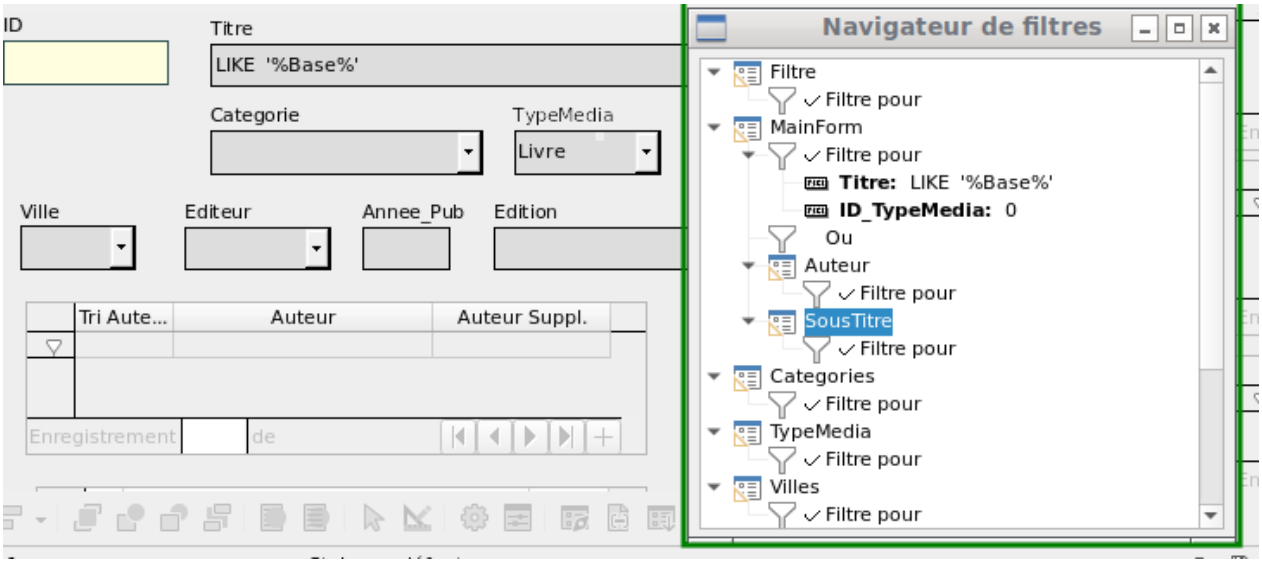
Le filtre basé sur un formulaire offre un formulaire pour insérer des valeurs réelles à utiliser pour le filtrage plutôt qu'un simple bouton de filtre. Ici, vous pouvez filtrer plusieurs valeurs en même temps. Ces valeurs peuvent être combinées sous forme de condition conjointe (AND) ou d'alternatives (OR).



Start Filtre Basé sur un formulaire

Les valeurs de filtre entrées ne doivent pas être exclusivement uniques. Ici donc vous pouvez formuler des conditions comme celles décrites dans le filtrage des tables. Ainsi, LIKE '% base%' saisi dans le champ du titre du média donnera tous les enregistrements dont le titre contient le mot 'base'.

Le filtrage des champs de zones de liste est effectué par sélection directe du contenu des champs de liste affichés. Vous n'avez pas besoin de saisir les clés étrangères sous-jacentes.

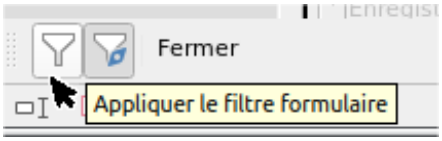


Ici, nous recherchons tous les enregistrements avec le TypeMedia "Livre" (correspondant à une valeur de clé étrangère de "0") et avons également quelque part dans le titre la séquence de caractères "Base". Lorsque vous entrez `LIKE '%Base%*'`, l'interface utilisateur graphique remplace les symboles génériques SQL habituels «%» par le caractère «*», un symbole générique plus familier pour la plupart des utilisateurs.

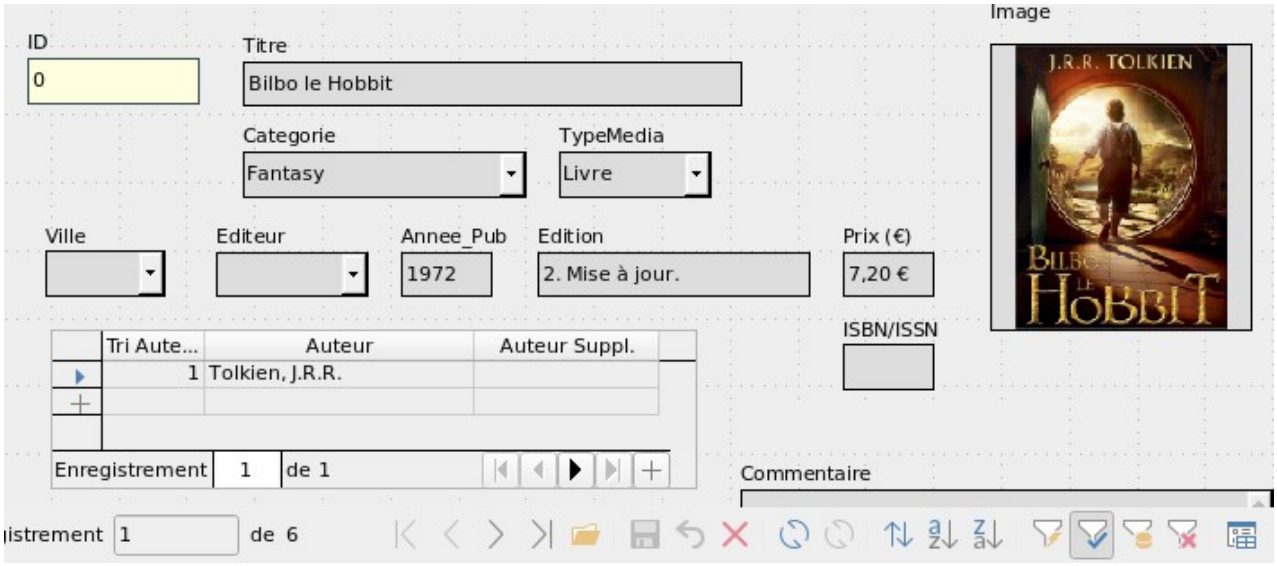
Nous pourrions également ajouter la condition que tous les enregistrements pour lesquels l'année de publication est "1973" soient affichés.

Lors de la saisie, le filtre basé sur un formulaire affiche uniquement les champs qui affichent une coche avant OR. De cette manière, plusieurs conditions peuvent être définies pour le champ. Malheureusement, cela provoque la suppression par le filtre de l'affichage du contenu du champ de liste TypeMedia, alors que cela fonctionne dans le contrôle de table du sous-formulaire.

Lors de la création du filtre, vous ne voyez pas la barre de navigation ; à la place, les choix du filtre basé sur le formulaire sont affichés. Vous filtrez selon vos besoins, insérez les valeurs de filtre, puis fermez simplement l'affichage.



Le filtre est appliqué et le formulaire est filtré selon vos spécifications.



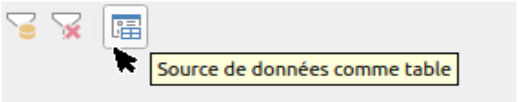
Lorsque le filtrage est activé, le nombre d'enregistrements est limité à ceux qui correspondent. Dans le cas, il y a six correspondances. Dans l'illustration, ni le titre ni le type de média ne correspondent ; la condition que l'année de publication soit inférieure à 1980 fournit la correspondance.

Ce filtre assez réussi présente encore l'inconvénient suivant :

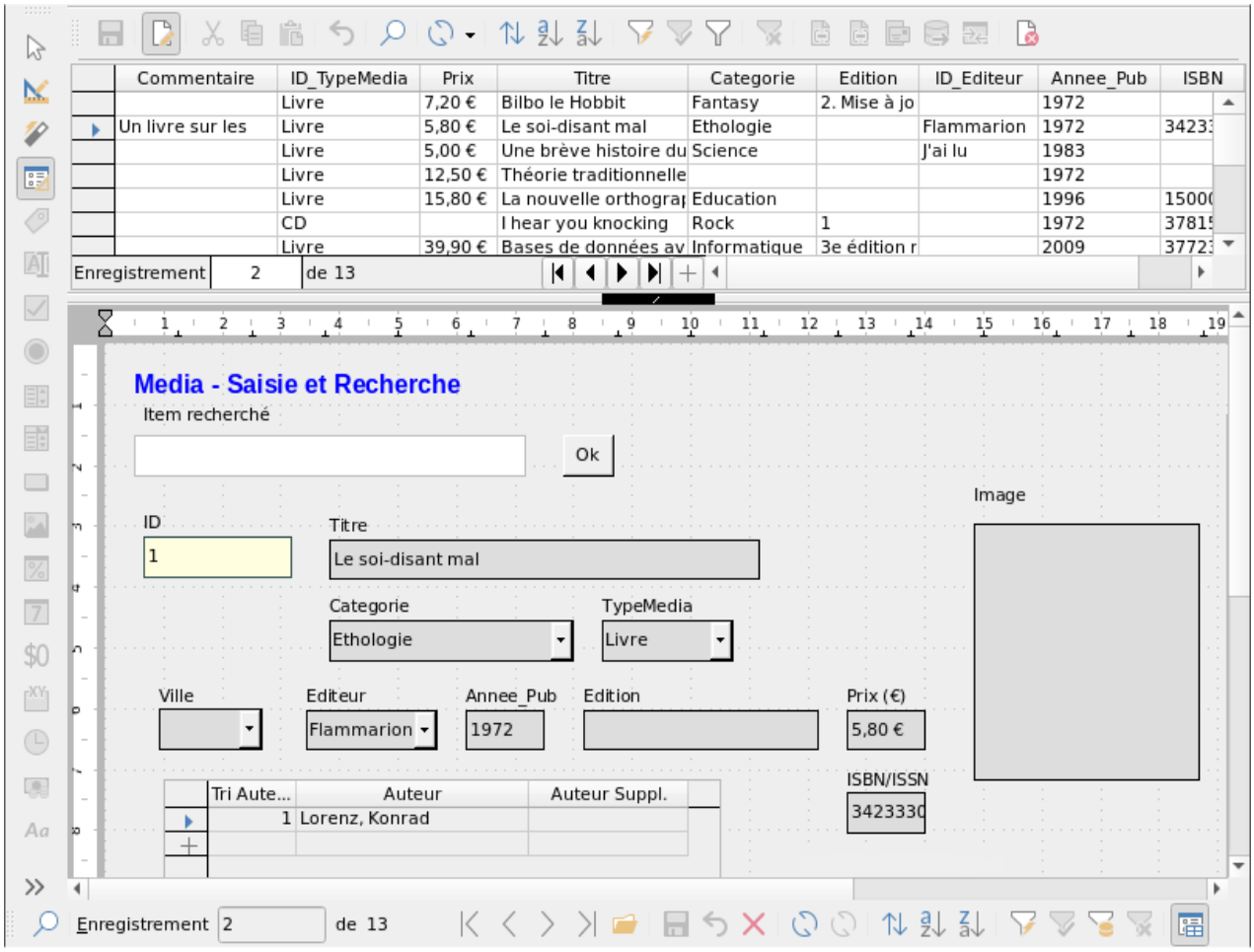
- Vous avez besoin de connaissances détaillées sur la façon de définir des conditions si vous recherchez quelque chose de plus compliqué que des champs complets. La plupart des utilisateurs, même s'ils sont habitués à utiliser les moteurs de recherche, n'ont pas les compétences nécessaires.

Filtrer avec le filtre par défaut

Le filtre par défaut peut être atteint en demandant l'affichage de la source de données sous forme de tableau. Ensuite, les procédures sont les mêmes que lors de l'utilisation du filtre par défaut dans une table.



Lancement du filtre par défaut avec la source de données sous forme de tableau



À droite du navigateur de Formulaire, vous pouvez voir un bouton *Source de données comme table*. En vue tableau, le filtre par défaut devient disponible. Contrairement à la vue de la table Medias réelle, les clés étrangères sont affichées, non pas avec leurs valeurs réelles, mais avec le contenu qui y correspond. Un clic sur le champ TypeMedia montre qu'il s'agit de la source du champ de liste qui apparaît dans le formulaire.

Note

Malheureusement, la vue ci-dessus montre un bogue qui remonte au début de Base. Le numéro ISBN à 13 chiffres n'est pas correctement formaté. Le champ correspondant reçoit à la place le nombre minimum de chiffres possible. Si un numéro a plus de 9 chiffres, attention : la modification peut entraîner une perte de données (bogue 82411).

Ce bogue apparaît également dans les contrôles de table. Ils s'affichent correctement si vous utilisez un champ formaté au lieu d'un champ numérique.

La vue tableau d'un formulaire présente les inconvénients suivants pour cette méthode de recherche :

- La recherche fonctionne uniquement dans la seule table ou requête qui sous-tend le formulaire et non dans un sous-formulaire qui lui appartient.
- Dans la vue des données, les champs de liste sont affichés, mais vous ne pouvez pas les utiliser pour le filtrage. Une fois de plus, vous devez connaître les valeurs de clé étrangère réelles pour ces champs.
  Par exemple, si le type de média est "Livre" et qu'il a la valeur de clé primaire "1", alors la

table Medias a “1” stocké dans ce champ. Vous devez rechercher cette valeur, même si la vue composite affiche la valeur “Livre”.

Résumé

Les fonctions de recherche et de filtrage sont d’une utilité limitée pour les formulaires. La recherche est beaucoup trop lente et ne limite pas le nombre total d’enregistrements à ceux qui correspondent vraiment. Les filtres ne fonctionnent que dans un formulaire à la fois. Si vous souhaitez filtrer dans un sous-formulaire, seul le sous-formulaire est affecté. Vous ne pouvez tout simplement pas obtenir le nombre correct d’enregistrements correspondants.

Par conséquent, dans les formulaires de la base de données d’exemple, une fonction de recherche conviviale a été intégrée dans le formulaire, une procédure qui doit être élaborée à la main et nécessite une certaine connaissance de SQL.

Saisie d’enregistrements et navigation

Si un formulaire a été conçu de telle sorte que son ouverture active le focus de contrôle, le curseur apparaîtra dans le premier champ de saisie. Au lieu d’utiliser la souris pour passer d’un champ à l’autre, vous utilisez la touche de tabulation pour passer au champ suivant dans la séquence.

Un formulaire vous permet de passer à un autre champ, soit avec la souris, soit avec un raccourci.

En utilisant **Propriétés : Zone de texte > Champ d’étiquette**, un champ d’étiquette est attribué. L’étiquette de ce champ d’étiquette passe de “Nom” à “No~m”. Dans le formulaire, la lettre “m” de l’étiquette est maintenant soulignée. Vous pouvez maintenant accéder instantanément à ce champ de texte en utilisant le raccourci Alt + m. Cela ne fonctionne que si le curseur est déjà dans un champ du formulaire.

En principe, n’importe quelle lettre peut être utilisée comme raccourci, car le saut a lieu entièrement dans le formulaire. Cependant, si le curseur n’est pas initialement dans un contrôle de formulaire, les raccourcis activent plutôt l’interface utilisateur de LibreOffice. Ainsi, par exemple, le raccourci “a” ne passerait pas au champ Nom mais ouvrirait à la place le menu Tableau.

Malheureusement, les sauts de champ ne fonctionnent que dans un seul formulaire, pas dans les structures qui impliquent des sous-formulaires. Un saut d’un sous-formulaire vers le formulaire principal n’est actuellement pas possible.

Lorsque le curseur est dans un contrôle de table, il sort lorsque vous utilisez *Ctrl+Tab*. Parfois, il arrive que cette combinaison de touches n’ait pas l’effet souhaité. Voici donc une alternative :[1]

Un bouton est créé dans le sous-formulaire. Son titre est “~Nouveau”. Le caractère tilde est utilisé, comme décrit ci-dessus, afin que *Alt+N* puisse être utilisé pour activer le bouton. Le champ Informations supplémentaires du bouton nomme le champ du formulaire principal vers lequel le curseur doit être transféré lors de l’activation. Le bouton est lié à la macro suivante à l’aide de **Propriétés > Événements > Exécuter l’action :**

```
SUB SautVersFormulairePrincipal(oEvent AS OBJECT)
    DIM oChamp AS OBJECT
    DIM oFormulaire AS OBJECT
    DIM oDoc AS OBJECT
    DIM oController AS OBJECT
```

1 Voir aussi l’exemple de base de données Exemple_Saut_Curseur_Formulaires.odb.associée à ce manuel.

```
    DIM oVue AS OBJECT
    DIM stRaccourci AS STRING
    oChamp = oEvent.Source.Model
    stRaccourci = Mid(oChamp.Label,InStr(oChamp.Label,"~") + 1, 1)
    oFormulaire = oChamp.Parent.Parent
    SELECT CASE stRaccourci
        CASE "n"
            oFormulaire.MoveToInsertRow()
        CASE "a"
            IF oFormulaire.isLast() THEN
                oFormulaire.MoveToInsertRow()
            ELSE
                oFormulaire.Next()
            END IF
    END SELECT
    oDoc = thisComponent
    oController = oDoc.getCurrentController()
    oVue = oController.getControl(oFormulaire.getByname(oChamp.Tag))
    oVue.setFocus
END SUB
```

La macro peut être utilisée simplement pour revenir au formulaire principal. Elle peut également être utilisée avec les paramètres ci-dessus pour créer un nouvel enregistrement immédiatement ou, à l'aide d'un autre bouton de la barre de navigation pour Enregistrement suivant, pour passer à l'enregistrement suivant. Si le curseur dans le formulaire principal est sur le dernier enregistrement, le saut conduira à un nouvel enregistrement. Pour plus d'informations sur les macros, reportez-vous au chapitre 9.

Passer à un bouton nécessite que le bouton soit visible. Cependant, il peut être assez petit ou même avoir une largeur et une hauteur de 0 cm. Si la largeur et la hauteur ne sont pas définies, il peut arriver que le bouton apparaisse comme une ligne parcourant tout l'écran.

Impression à partir de formulaires

Les formulaires peuvent être construits de telle sorte qu'une impression directe du formulaire soit possible. Pour obtenir des résultats utilisables, vous devez veiller à ne pas placer d'éléments en dehors de la zone imprimable. Choisissez **Affichage > Normal**. Les propriétés de la page peuvent ensuite être définies. Un fond coloré n'est pas un avantage ici.

Tout élément individuel peut être exclu de l'impression en utilisant **Propriétés > Général > Imprimable.**

Si la base de données est enregistrée dans LibreOffice (en utilisant **Outils > Options > LibreOffice Base > Base de données**, ou directement dans le cadre du processus de création), le formulaire peut être utilisé pour les publipostages. Le formulaire est ouvert pour modification. Les sources de données sont rendues accessibles en utilisant **Affichage > Sources de données** ou en appuyant sur *F4*. Les champs de base de données peuvent maintenant être glissés dans le formulaire par leurs en-têtes de tableau. Ensuite, le formulaire est enregistré. Si le même formulaire est maintenant ouvert pour la saisie, LibreOffice reconnaît qu'il s'agit de champs pour un publipostage et vous demande si vous souhaitez imprimer les lettres.

Vous trouverez des détails sur la façon de procéder à un publipostage dans le Chapitre 7, Liaison aux bases de données.