



Guide Base

Appendice A
Documentations diverses

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

Jenna Sargent

Des éditions précédentes

Pulkit Krishna

Jean Hollis Weber

Randolph GAMO

Jochen Schiffers

Robert Großkopf

Jost Lange

Hazel Russman

Jean Hollis Weber

Traduction

Jean-Michel COSTE

Relecteurs

Patrick Auclair

Retour d'information

Veillez 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
Codes à barres	4
Types de données pour l'éditeur de tables	4
Entiers.....	4
Nombres à virgule flottante.....	4
Autres.....	5
Date/Temps/Heures.....	5
Autres.....	5
Types de données dans StarBasic	6
Nombres.....	6
Others.....	6
Fonctions intégrées et procédures stockées	7
Numeriques.....	8
Texte.....	9
Date/Heure.....	11
Connexion à la base de données.....	12
Système.....	14
Caractères de contrôle à utiliser dans les requêtes	15
Quelques commandes uno à utiliser avec un bouton	15
Tables d'informations pour HSQLDB	16
Réparation de base de données pour les fichiers *.odb	17
Récupération du fichier d'archive de la base de données.....	17
Informations supplémentaires sur les fichiers d'archive de base de données.....	19
Gestion de la base de données interne Firebird	26
Rendre AutoChamp disponible.....	26

Codes à barres

Pour pouvoir utiliser la fonction d'impression de codes-barres, la police ean13.ttf doit être installée. Cette police est disponible gratuitement <http://www.codebarre.be/barcodefont/ean13.ttf>.

Les codes-barres EAN13 peuvent être créés en utilisant ean13.ttf comme suit :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Nombre	Majuscule, A=0 B=1, etc.						*	Minuscule, a=0 b=1, etc.						+

Voir aussi la requête Barcode_EAN13_ttf_Rapport et le rapport correspondant dans l'exemple de base de données Media_sans_Macros.

Code-barres pour étiquetage



Figure 1: Impression du rapport avec le code ean13

Types de données pour l'éditeur de tables

Entiers				
Type	Option	HSQLDB	Intervalle	Stockage
Tiny Integer	TINYINT	TINYINT	$2^8 = 256$ - 128 to + 127	1 Byte
Small Integer	SMALLINT	SMALLINT	$2^{16} = 65536$ - 32768 to + 32767	2 Octets
Integer	INTEGER	INTEGER INT	$2^{32} = 4294967296$ - 2147483648 to + 2147483647	4 Octets
BigInt	BIGINT	BIGINT	2^{64}	8 Octets
Nombres à virgule flottante				
Type	Option	HSQLDB	Intervalle	Stockage
Decimal	DECIMAL	DECIMAL	Illimité, jusqu'à 50 emplacements dans l'interface graphique, virgule décimale fixe, précision parfaite	variable

Nombre	NUMERIC	NUMERIC	Illimité, jusqu'à 50 emplacements dans l'interface graphique, virgule décimale fixe, précision parfaite	variable
Flottant	FLOAT	(DOUBLE used instead)		
Réel	REAL	REAL		
Double	DOUBLE	DOUBLE [PRECISION] FLOAT	Réglable, pas exact, 15 décimales maximum	8 Octets

Autres

Type	Option	HSQLDB	Intervalle	Stockage
Texte	VARCHAR	VARCHAR	Ajustable	variable
Texte	VARCHAR_IGNORECASE	VARCHAR_IGNORECASE	Réglable, la plage affecte le tri	variable
Texte (fixe)	CHAR	CHAR CHARACTER	Réglable, reste du texte réel remplacé par des espaces	fixe
Memo	LONGVARCHAR	LONGVARCHAR		variable

Date/Temps/Heures

Type	Option	HSQLDB	Intervalle	Stockage
Date	DATE	DATE		4 Octets
Time	TIME	TIME		4 Octets
Date/Time	TIMESTAMP	TIMESTAMP DATETIME	Réglable (0,6 – 6 signifie avec des millisecondes)	8 Octets

Autres

Type	Option	HSQLDB	Range	Storage space
Oui/Non	BOOLEAN	BOOLEAN BIT		
Champ binaire (fixe)	BINARY	BINARY	Comme Integer	fixe

Autres				
Champ binaire	VARBINARY	VARBINARY	Comme Integer	variable
Image	LONGVARBINARY	LONGVARBINARY	Comme Integer	variable, destiné aux images plus grandes
OTHER	OTHER	OTHER OBJECT		

Dans les définitions de table, et lorsque les types de données sont modifiés dans les requêtes utilisant les fonctions « convertir » ou « cast », certains types de données attendent des informations sur le nombre de caractères (a), la précision (g, correspondant au nombre total de caractères) et le nombre de décimales (d). Les types sont CHAR (a), VARCHAR (a), DOUBLE (g), NUMERIC (g, d), DECIMAL (g, d) et TIMESTAMP (g).

TIMESTAMP (g) ne peut avoir que deux valeurs : « 0 » et « 6 ». « 0 » signifie qu'aucune seconde ne sera stockée dans la partie décimale (dixièmes, centièmes...). La précision des horodatages ne peut être donnée que directement à l'aide de commandes SQL. Donc, si vous stockez les horaires d'un type de sport, vous devez définir TIMESTAMP (6) en utilisant **Outils> SQL** à l'avance.

Types de données dans StarBasic

Nombres				
Type	Correspond à HSQLDB	Valeur initiale	Remarques	Exigences de stockage
Entier	SMALLINT	0	$2^{16} = - 32768$ bis $+ 32767$	2 Octets
Long	INTEGER	0	$2^{32} =$ $- 2147483648$ bis $+ 2147483647$	4 Octets
Simple		0.0	Decimal:.	4 Octets
Double	DOUBLE	0.0	Decimal:.	8 Octets
Monétaire	Resembles DECIMAL, NUMERIC	0.0000	4 décimales fixes	8 Octets
Autres				
Type	Corresponds to HSQLDB	Initial value	Remarks	Storage requirements
Boolean	BOOLEAN	False	1 = oui, tout le reste : non.	1 Octets
Date	TIMESTAMP	00:00:00	Jusqu'à 65536 caractères	8 Octets
String	VARCHAR	Chaine vide		variable
Object	OTHER	Null		variable

Variant		Empty	Peut accepter n'importe quel (autre) type de données	variable
---------	--	-------	--	----------

Il existe de grands risques dans la conversion de données, en particulier avec des valeurs numériques. Par exemple, les clés primaires des bases de données sont le plus souvent du type INTEGER. Si ceux-ci sont lus par une macro, la variable dans laquelle ils sont stockés doit être de type Long, car sa taille correspond au type INTEGER dans Base. L'instruction de lecture correspondante est getLong.

Fonctions intégrées et procédures stockées

Les fonctions suivantes sont disponibles dans la HSQLDB intégrée. Malheureusement, une ou deux fonctions ne peuvent être utilisées que lorsque la commande **Exécuter SQL** directement est choisie. Cela empêchera alors ces requêtes d'être mémorisées et modifiées.

Les fonctions qui fonctionnent avec l'interface utilisateur graphique sont marquées [Fonctionne dans l'IGU] (*Interface Graphique Utilisateur*). Les fonctions qui ne fonctionnent que dans les commandes SQL directes sont marquées [SQL direct – ne fonctionne pas dans l'IGU].

Numeriques

Comme nous traitons ici de nombres à virgule flottante, assurez-vous de faire attention aux paramètres des champs dans les requêtes. La plupart du temps, l'affichage des décimales est limité, de sorte que dans certains cas, il peut y avoir des résultats inattendus. Par exemple, la colonne 1 peut afficher 0,00 mais contient en réalité 0,001 et la colonne 2, 1000. Si la colonne 3 est définie pour afficher Colonne 1 * Colonne 2, elle affichera en fait 1.

ABS(d)	Renvoie la valeur absolue d'un nombre. [Fonctionne dans l'IGU]
ACOS(d)	Renvoie l'arc cosinus. [Fonctionne dans l'IGU]
ASIN(d)	Returns the arcsine. [Fonctionne dans l'IGU]
ATAN(d)	Renvoie l'arc tangente. [Fonctionne dans l'IGU]
ATAN2(a, b)	Renvoie l'arc tangente à l'aide de coordonnées, où a est la valeur de l'axe x, b la valeur de l'axe y. [Fonctionne dans l'IGU]
BITAND(a, b)	La forme binaire de a et la forme binaire de b doivent avoir 1 à la même position pour donner 1 dans le résultat. BITAND (3,5) donne 1 ; 0011 ET 0101 = 0001 [Fonctionne dans l'IGU]
BITOR(a, b)	La forme binaire de a ou la forme binaire de b doit avoir 1 à la même position pour donner 1 dans le résultat. BITOR (3,5) donne 7 ; 0011 OU 0101 = 0111 [Fonctionne dans l'IGU]
CEILING(d)	Renvoie le plus petit nombre entier qui n'est pas inférieur à d. [Fonctionne dans l'IGU]
COS(d)	Renvoie le cosinus. [Fonctionne dans l'IGU]
COT(d)	Renvoie la cotangente. [Fonctionne dans l'IGU]
DEGREES(d)	Convertit les radians en degrés. [Fonctionne dans l'IGU]
EXP(d)	Renvoie e^d (e : (2.718...)). [Fonctionne dans l'IGU]
FLOOR(d)	Renvoie le plus grand nombre entier qui n'est pas supérieur à d. [Fonctionne dans l'IGU]
LOG(d)	Renvoie le logarithme naturel en base e. [Fonctionne dans l'IGU]
LOG10(d)	Renvoie le logarithme en base 10. [Fonctionne dans l'IGU]

MOD(a, b)	Renvoie le reste sous forme de nombre entier, dans la division de 2 nombres entiers. MOD (11,3) renvoie 2, car $3 * 3 + 2 = 11$ [Fonctionne dans l'IGU]
PI()	Renvoie π (3,1415...). [Fonctionne dans l'IGU]
POWER(a, b)	a^b , POWER(2,3) = 8, car $2^3 = 8$ [Fonctionne dans l'IGU]
RADIANS(d)	Convertit les degrés en radians. [Fonctionne dans l'IGU]
RAND()	Renvoie un nombre aléatoire supérieur ou égal à 0,0 et inférieur à 1,0. [Fonctionne dans l'IGU]
ROUND(a, b)	Arrondit a à b décimales. [Fonctionne dans l'IGU]
ROUNDMAGIC(d)	Résout les problèmes d'arrondi résultant de l'utilisation de nombres à virgule flottante. 3.11-3.1-0.01 n'est pas exactement 0, mais est affiché comme 0 dans l'interface graphique. ROUNDMAGIC en fait une valeur zéro réelle. [Fonctionne dans l'IGU]
SIGN(d)	Renvoie -1 si d est inférieur à 0, 0 si d est égal à 0 et 1 si d est supérieur à 0. [Fonctionne dans l'IGU]
SIN(A)	Renvoie le sinus d'un angle en radians. [Fonctionne dans l'IGU]
SQRT(d)	Renvoie la racine carrée. [Fonctionne dans l'IGU]
TAN(A)	Renvoie la tangente d'un angle en radians. [Fonctionne dans l'IGU]
TRUNCATE(a, b)	Truncates a to b decimal places. TRUNCATE(2.37456,2) = 2.37 [Fonctionne dans l'IGU]
Texte	
ASCII(s)	Renvoie le code ASCII de la première lettre de la chaîne. [Fonctionne dans l'IGU]
BIT_LENGTH(str)	Renvoie la longueur de la chaîne de texte str en bits. [Fonctionne dans l'IGU]
CHAR(c)	Returns the letter corresponding to the ASCII code c. [Fonctionne dans l'IGU]
CHAR_LENGTH(str)	Renvoie la longueur de la chaîne str en caractères. [Fonctionne dans l'IGU]

CONCAT(str1, str2)	Concatène str1 et str2. [Fonctionne dans l'IGU]
'str1' 'str2' 'str3' or 'str1'+ 'str2'+ 'str3'	Concatenates str1, str2, and str3. A simpler alternative to CONCAT. [Fonctionne dans l'IGU]
DIFFERENCE(s1, s2)	Renvoie la différence de son entre s1 et s2. Seul un nombre entier est émis. 0 signifie qu'ils sonnent de la même manière. Par exemple, « sert » et « serre » donnent 0, « pair » et « faire » donnent 1, « mer » et « mère » donnent 0. [Fonctionne dans l'IGU]
HEXTORAW(s1)	Traduit le code hexadécimal en d'autres caractères. [Fonctionne dans l'IGU]
INSERT(s, start, len, s2)	Renvoie une chaîne de texte, avec une partie du texte remplacée. En commençant à d une longueur l est découpée dans le texte s et remplacée par le texte s2. INSERT (« Consolation », 3, 4, firm) convertit Consolation en Confirmation, où la longueur du texte inséré peut être supérieure à celle du texte supprimé sans causer de problèmes. Donc INSERT (« Bundesbahn », 3, 5, s et B) donne « Bus und Bahn ». [Fonctionne dans l'IGU]
LCASE(s)	Convertit une chaîne en minuscules. [Fonctionne dans l'IGU]
LEFT(s, compte)	Renvoie le nombre de caractères spécifié par compte à partir du début de la chaîne s. [Fonctionne dans l'IGU]
LENGTH(s)	Renvoie la longueur de la chaîne s en caractères. [Fonctionne dans l'IGU]
LOCATE(cherche, s, [début])	Renvoie la première correspondance du terme recherché dans la chaîne s. La correspondance est donnée sous forme de nombre de décalage : (1 = gauche, 0 = introuvable). La définition d'un point de départ dans la chaîne de texte est facultative. [Fonctionne dans l'IGU]
LTRIM(s)	Supprime les espaces de début et les caractères non imprimables au début d'une chaîne de texte. [Fonctionne dans l'IGU]
OCTET_LENGTH(str)	Renvoie la longueur d'une chaîne de texte en octets. Cela correspond à deux fois la longueur de la chaîne. [Fonctionne dans l'IGU]
RAWTOHEX(s1)	Convertit en hexadécimaux, inverse deHEXTORAW(). [Fonctionne dans l'IGU]
REPEAT(s, count)	Répète le nombre de fois de la chaîne de texte. [Fonctionne dans l'IGU]

REPLACE(s, remplace, s2)	Remplace toutes les occurrences de remplace existantes dans la chaîne de texte s par la chaîne s2. [Fonctionne dans l'IGU]
RIGHT(s, count)	Opposé de LEFT ; renvoie le dernier nombre de caractères à la fin d'une chaîne de texte. [Fonctionne dans l'IGU]
RTRIM(s)	Supprime tous les espaces et les caractères non imprimables à la fin d'une chaîne de texte. [Fonctionne dans l'IGU]
SOUNDEX(s)	Renvoie un code à 4 caractères, correspondant au son de s. Correspond à la fonction DIFFÉRENCE(). [Fonctionne dans l'IGU]
SPACE(count)	Renvoie le nombre d'espaces. [Fonctionne dans l'IGU]
SUBSTR(s, start[, len])	Abréviation de SUBSTRING. [Fonctionne dans l'IGU]
SUBSTRING(s, début[, longueur])	Renvoie le texte s à partir de la position de début (1 = gauche). Si longueur est omis, la chaîne entière est renvoyée. [Fonctionne dans l'IGU]
UCASE(s)	Convertit une chaîne en majuscules. [Fonctionne dans l'IGU]
LOWER(s)	Comme LCASE(s). [Fonctionne dans l'IGU]
UPPER(s)	Comme UCASE(s). [Fonctionne dans l'IGU]
Date/Heure	
CURDATE()	Renvoie la date actuelle. [Fonctionne dans l'IGU]
CURTIME()	Renvoie l'heure actuelle. [Fonctionne dans l'IGU]
DATEDIFF(string, datetime1, datetime2)	Différence de date entre deux dates – compare les valeurs de date / heure. L'entrée en chaîne détermine les unités dans lesquelles la différence est renvoyée : ms = milliseconde, ss = seconde, mi = minute, hh = heure, jj = jour, mm = mois, aa = année. Les formes longues et courtes peuvent être utilisées pour la chaîne. [Fonctionne dans l'IGU]
DAY(date)	Renvoie le jour du mois (1-31). [Fonctionne dans l'IGU]
DAYNAME(date)	Renvoie le nom anglais du jour. [Fonctionne dans l'IGU]

DAYOFMONTH(date)	Renvoie le jour du mois (1-31). Synonyme de DAY(). [Fonctionne dans l'IGU]
DAYOFWEEK(date)	Renvoie le jour de la semaine sous forme de nombre (1 représente dimanche). [Fonctionne dans l'IGU]
DAYOFYEAR(date)	Renvoie le jour de l'année (1-366). [Fonctionne dans l'IGU]
HOUR(time)	Renvoie l'heure (0-23). [Fonctionne dans l'IGU]
MINUTE(time)	Renvoie les minutes (0-59). [Fonctionne dans l'IGU]
MONTH(date)	Renvoie le mois (1-12). [Fonctionne dans l'IGU]
MONTHNAME(date)	Renvoie le nom anglais du mois. [Fonctionne dans l'IGU]
NOW()	Renvoie la date actuelle et l'heure actuelle ensemble sous forme d'horodatage. Vous pouvez également utiliser CURRENT_TIMESTAMP. [Fonctionne dans l'IGU]
QUARTER(date)	Renvoie le trimestre de l'année (1-4). [Fonctionne dans l'IGU]
SECOND(time)	Renvoie la partie des secondes du temps (0-59). [Fonctionne dans l'IGU]
WEEK(date)	Renvoie la semaine de l'année (1-53). [Fonctionne dans l'IGU]
YEAR(date)	Renvoie la partie année d'une entrée de date. [Fonctionne dans l'IGU]
CURRENT_DATE	Synonyme de CURDATE(), SQL-Standard. [Fonctionne dans l'IGU]
CURRENT_TIME	Synonyme de CURTIME(), SQL-Standard. [Fonctionne dans l'IGU]
CURRENT_TIMESTAMP	Synonyme de NOW(), SQL-Standard. [Fonctionne dans l'IGU]

Connexion à la base de données

À l'exception de IDENTITY(), qui n'a aucune signification dans Base, tout cela peut être effectué à l'aide d'une commande SQL directe.

DATABASE()	Renvoie le nom de la base de données à laquelle appartient cette connexion. [Fonctionne dans l'IGU]
USER()	Renvoie le nom d'utilisateur de cette connexion. [SQL Direct– ne fonctionne pas dans l'IGU]

CURRENT_USER	Fonction standard SQL, synonyme de USER (). [Fonctionne dans l'IGU]
IDENTITY()	Renvoie la dernière valeur d'un champ de valeur automatique, qui a été créé dans la connexion actuelle. Ceci est utilisé dans le codage de macro pour transférer une clé primaire dans une table pour devenir une clé étrangère pour une autre table. [Fonctionne dans l'IGU]

Systeme	
IFNULL(exp, valeur)	Si exp est NULL, la valeur est retournée, sinon exp est retournée. Comme extension, COALESCE() peut être utilisée. Exp et valeur doivent avoir le même type de données. [Fonctionne dans l'IGU]
CASEWHEN(exp, v1, v2)	Si exp est vrai, v1 est renvoyé, sinon v2. Alternativement, CASE WHEN peut aussi être utilisé. CASE WHEN fonctionne mieux avec l'interface graphique. [Fonctionne dans l'IGU]
CONVERT(terme, type)	Convertit le terme en un autre type de données. [Fonctionne dans l'IGU]
CAST(terme AS type)	Synonyme de CONVERT(). [Fonctionne dans l'IGU]
COALESCE(expr1, expr2, expr3,...)	Si expr1 n'est pas NULL, renvoie expr1, sinon expr2 est vérifié, puis expr3 et ainsi de suite. [Fonctionne dans l'IGU]
NULLIF(v1, v2)	Si v1 est égal à v2, null est renvoyé, sinon la valeur de v1 est renvoyée. Les données doivent être de type comparable. [Fonctionne dans l'IGU]
CASE v1 WHEN v2 THEN v3 [ELSE v4] END	Si v1 est égal à v2, v3 est renvoyé. Sinon, v4 est retourné ou NULL, s'il n'y a pas de condition ELSE. [SQL Direct– ne fonctionne pas dans l'IGU]
CASE WHEN expr1 THEN v1 [WHEN expr2 THEN v2] [ELSE v4] END	If expr1 is true, v1 is returned [optionally further conditions can be set]. Otherwise v4 is returned or NULL if there is no ELSE condition. [Fonctionne dans l'IGU]
EXTRACT ({YEAR MONTH DAY HOUR MINUTE SECOND} FROM <date ou heure>)	Peut remplacer de nombreuses fonctions de date et d'heure. Renvoie l'année, le mois, le jour, etc. à partir d'une valeur de date ou de date / heure. [Fonctionne dans l'IGU]
POSITION(<string expression> IN <string expression>)	Si la première chaîne est contenue dans la seconde, le décalage de la première chaîne est donné, sinon 0 est renvoyé. [Fonctionne dans l'IGU]
SUBSTRING(<expression chaine> FROM <expression numerique> [FOR <numeric expression>])	Renvoie une partie d'une chaîne de texte à partir de la position spécifiée dans FROM, éventuellement jusqu'à la longueur indiquée dans FOR. [Fonctionne dans l'IGU]
TRIM({LEADING TRAILING BOTH}) FROM < expression chaine>)	Les caractères spéciaux et les espaces non imprimables sont supprimés. [Fonctionne dans l'IGU]

Caractères de contrôle à utiliser dans les requêtes

Les champs peuvent être liés entre eux dans les requêtes. Deux champs dans

```
SELECT "Prenom", "Nom" FROM "Table"
```

devient un champ unique en utilisant :

```
SELECT "Prenom" || ' ' || "Nom" FROM "Table"
```

Ici, un espace supplémentaire est inséré. Cela peut être n'importe quel caractère ; tant qu'il est entouré de " , il sera interprété comme du texte. Parfois, cependant, il est nécessaire d'insérer des caractères non imprimables tels que des fin de lignes, par exemple lors de la préparation de rapports. Voici une courte liste de caractères de contrôle. Plus d'informations à ce sujet sont disponibles à l'adresse https://fr.wikipedia.org/wiki/Caract%C3%A8re_de_contr%C3%B4le.

CHAR(9)	Tabulation horizontale	Positionne le caractère suivant dans une colonne de type <i>tab stop</i> (tabulation).
CHAR(10)	Line feed (saut de ligne)	Dans les lettres de publipostage et le Générateur de Rapports, crée un saut de ligne (Linux, Unix, Mac).
CHAR(13)	Carriage return (retour chariot)	Saut de ligne lorsqu'il est combiné avec le retour chariot dans Windows CHAR (13) CHAR (10). Peut également être utilisé sous Linux et Mac, d'où la variante universelle.



Note

La combinaison "Ligne suivante – Retour chariot" est un héritage de la machine à écrire manuelle, puis des terminaux télétype et certaines anciennes imprimantes à aiguille, avec lesquels deux commandes étaient nécessaires pour positionner la tête d'écriture à la ligne suivante puis en début de ligne. A sa création, le système Linux qui ne gérait pas ce type de terminal, n'utilise qu'un seul caractère pour le saut de ligne dans les fichiers texte.

Quelques commandes uno à utiliser avec un bouton

Un bouton peut être lié (par sa propriété Action) à diverses commandes uno. Pour cela, vous devez choisir **Propriétés : Bouton > Action > Ouvrir un document / une page Web**, puis par exemple l'**URL > uno : RecSearch** pour ouvrir la fonction de recherche. Souvent, vous devrez choisir **Focus sur Clic > Non** si l'action accède directement à un autre contrôle d'une manière qui nécessite qu'il ait le focus, par exemple **uno: Paste**, qui peut insérer le contenu du presse-papiers dans un contrôle texte.

La liste suivante ne contient que quelques commandes. Toutes les commandes de la barre d'outils de navigation sont déjà utilisables dans le bouton, mais elles peuvent également être créées à l'aide des commandes uno. De nombreuses commandes peuvent être découvertes à l'aide de l'enregistreur de macros, qui utilise souvent un "dispatcher" pour y accéder.

Uno-Command	Used for...
.uno:RecSearch	Ouvre la fonction de recherche dans un formulaire.
.uno:Paste	Coller à partir du presse-papiers. Fonctionne uniquement pour Focus sur clic > Non .
.uno:Copy	Copie le contenu sélectionné dans le presse-papiers. Fonctionne uniquement pour Focus sur clic > Non .
.uno:Print	Ouvre la boîte de dialogue d'impression du formulaire.

.uno:PrintDefault	Imprime avec l'imprimante par défaut sans afficher de boîte de dialogue.
-------------------	--

Tables d'informations pour HSQLDB

Dans une base de données, les informations sur toutes les propriétés de table et leurs connexions entre elles sont stockées dans la zone *INFORMATION_SCHEMA*. Ces informations permettent de créer des macros BASE qui nécessitent très peu d'arguments pour leurs procédures. Une application est donnée dans l'exemple de base de données du module Maintenance – la procédure *Purge_Table* pour le contrôle des dialogues.

Dans une requête, des informations individuelles et tous les champs qui en font partie peuvent être fournis de la manière suivante :

```
SELECT * FROM "INFORMATION_SCHEMA"."SYSTEM_ALIASES"
```

Contrairement à une table normale, il est ici nécessaire d'utiliser *INFORMATION_SCHEMA* comme préfixe du nom approprié dans la liste suivante :

```
SYSTEM_ALIASES
SYSTEM_ALLTYPEINFO
SYSTEM_BESTROWIDENTIFIER
SYSTEM_CACHEINFO
SYSTEM_CATALOGS
SYSTEM_CHECK_COLUMN_USAGE
SYSTEM_CHECK_CONSTRAINTS
SYSTEM_CHECK_ROUTINE_USAGE
SYSTEM_CHECK_TABLE_USAGE
SYSTEM_CLASSPRIVILEGES
SYSTEM_COLUMNPRIVILEGES
SYSTEM_COLUMNS
SYSTEM_CROSSREFERENCE
SYSTEM_INDEXINFO
SYSTEM_PRIMARYKEYS
SYSTEM_PROCEDURECOLUMNS
SYSTEM_PROCEDURES
SYSTEM_PROPERTIES
SYSTEM_SCHEMAS
SYSTEM_SEQUENCES
SYSTEM_SESSIONINFO
SYSTEM_SESSIONS
SYSTEM_SUPERTABLES
SYSTEM_SUPERTYPES
SYSTEM_TABLEPRIVILEGES
SYSTEM_TABLES
SYSTEM_TABLETYPES
SYSTEM_TABLE_CONSTRAINTS
SYSTEM_TEXTTABLES
SYSTEM_TRIGGERCOLUMNS
SYSTEM_TRIGGERS
SYSTEM_TYPEINFO
SYSTEM_UDTATTRIBUTES
SYSTEM_UDTS
SYSTEM_USAGE_PRIVILEGES
SYSTEM_USERS
SYSTEM_VERSIONCOLUMNS
SYSTEM_VIEWS
SYSTEM_VIEW_COLUMN_USAGE
SYSTEM_VIEW_ROUTINE_USAGE
SYSTEM_VIEW_TABLE_USAGE
```


La requête suivante donne un aperçu complet de toutes les tables de la base de données avec les types de champs, les clés primaires et les clés étrangères :

```
SELECT
"A"."TABLE_NAME",
"A"."COLUMN_NAME",
"A"."TYPE_NAME",
"A"."NULLABLE",
"B"."KEY_SEQ" AS "PRIMARYKEY",
"C"."PKTABLE_NAME" || '.' || "C"."PKCOLUMN_NAME" AS "FOREIGNKEY FOR"
FROM "INFORMATION_SCHEMA"."SYSTEM_COLUMNS" AS "A"
LEFT JOIN "INFORMATION_SCHEMA"."SYSTEM_PRIMARYKEYS" AS "B"
ON ("B"."TABLE_NAME" = "A"."TABLE_NAME" AND "B"."COLUMN_NAME" =
"A"."COLUMN_NAME")
LEFT JOIN "INFORMATION_SCHEMA"."SYSTEM_CROSSREFERENCE" AS "C"
ON ("C"."FKTABLE_NAME" = "A"."TABLE_NAME" AND "C"."FKCOLUMN_NAME" =
"A"."COLUMN_NAME")
WHERE "A"."TABLE_SCHEM" = 'PUBLIC'
```

Réparation de base de données pour les fichiers *.odb

Les sauvegardes régulières des données doivent être une pratique courante lors de l'utilisation d'un PC. Les copies de sauvegarde sont le moyen le plus simple de revenir à un état actuel, même à mi-chemin, pour vos données. Cependant, dans la pratique, cela fait souvent défaut.



Conseil

Pour effectuer des sauvegardes régulières de vos documents dans des états successifs, vous pouvez utiliser l'extension de [Sauvegarde avec incrémentation automatique](#).

Les formulaires, requêtes et rapports peuvent toujours être copiés à l'aide du presse-papiers dans une nouvelle base de données, à condition qu'une version précédente de la base de données ait été enregistrée. Mais si, pour une raison quelconque, la base de données actuelle ne peut plus être ouverte, le problème principal devient l'accès aux données.

En cas de panne soudaine du PC, il peut arriver que des bases de données ouvertes (bases de données internes HSQLDB ou Firebird) ne puissent plus être ouvertes dans LibreOffice. Au lieu de cela, lorsque vous essayez d'ouvrir la base de données, il vous est demandé un filtre correspondant au format.

Le problème ici est qu'une partie des données d'une base de données ouverte est contenue dans la mémoire de travail et n'est copiée que temporairement dans le stockage intermédiaire. Ce n'est que lorsque le fichier est fermé que toute la base de données est réécrite dans le fichier et reconditionnée.

Récupération du fichier d'archive de la base de données

Pour avoir à nouveau accès à vos données, la procédure suivante peut vous être utile :

- 1) Créez une copie de votre base de données pour les étapes suivantes.
- 2) Essayez d'ouvrir la copie avec un programme d'archivage. Dans le cas des fichiers *.odb, nous avons affaire à un format compressé, une archive Zip. Si le fichier ne peut pas être ouvert directement, essayez de le renommer de *.odb en *.zip. Si cela ne l'ouvre pas, votre base de données est en cours d'enregistrement (voir s'il existe un fichier *.lck au même nom dans le dossier de la base).
- 3) Les dossiers visibles sur la Figure 2 seront toujours visibles après l'ouverture d'un fichier de base de données dans un programme d'archivage.

- 4) Le fichier de base de données doit être décompressé. Les informations les plus importantes, en ce qui concerne les données, se trouvent dans la base de données des sous-dossiers dans les fichiers **data** et **script**.

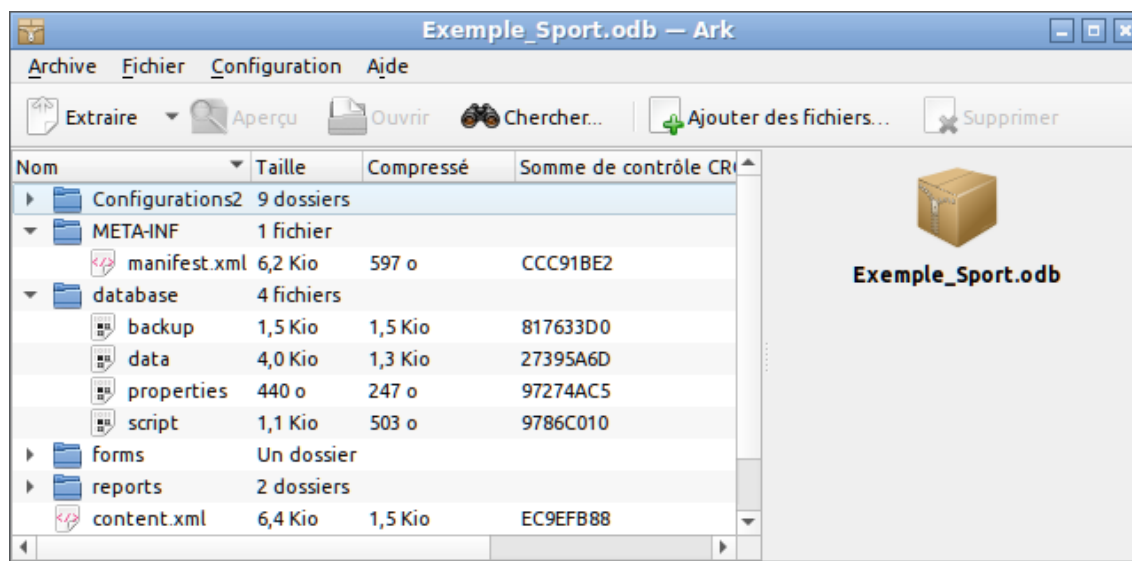


Figure 2: Vue de la structure de l'archive de base de données

- 5) Il peut être nécessaire de regarder le fichier de script et de le tester pour les contradictions. Cette étape peut cependant être laissée pour la phase de test. Le fichier script contient avant tout la description de la structure des tables.

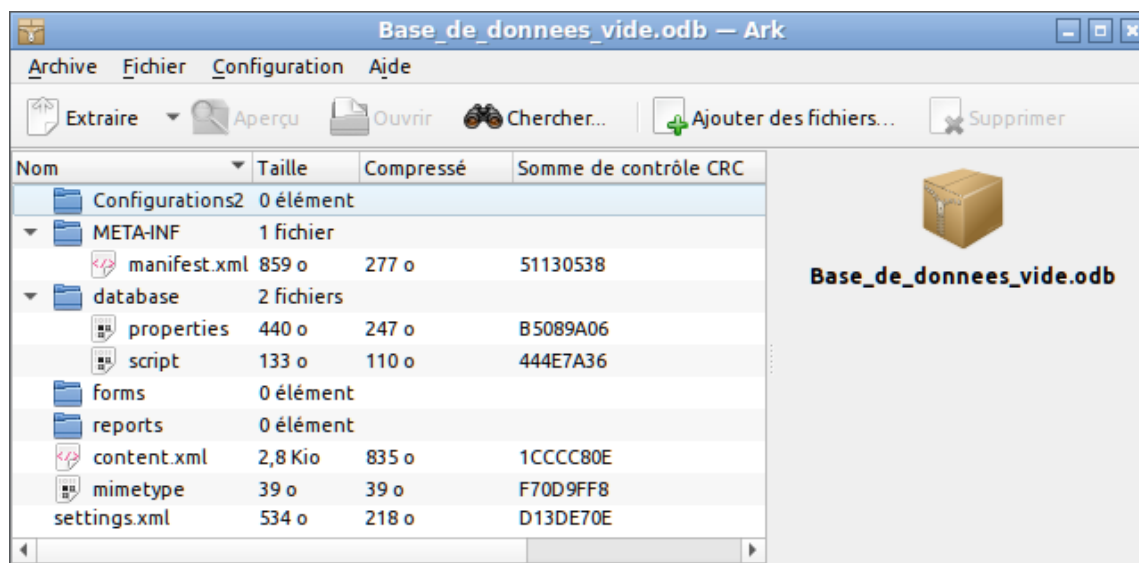


Figure 3: Vue de l'archive de la base vide

- 6) Créez un nouveau fichier de base de données vide et ouvrez ce fichier avec le programme d'archivage.
- 7) Remplacez les fichiers **data** et **script** dans le nouveau fichier de base de données par les fichiers décompressés à l'étape 4.
- 8) Fermez le programme d'archivage. S'il était nécessaire de renommer le fichier en *.zip avant de l'ouvrir dans le programme d'archivage (cela dépend de votre système d'exploitation), renommez-le maintenant à nouveau en *.odb.

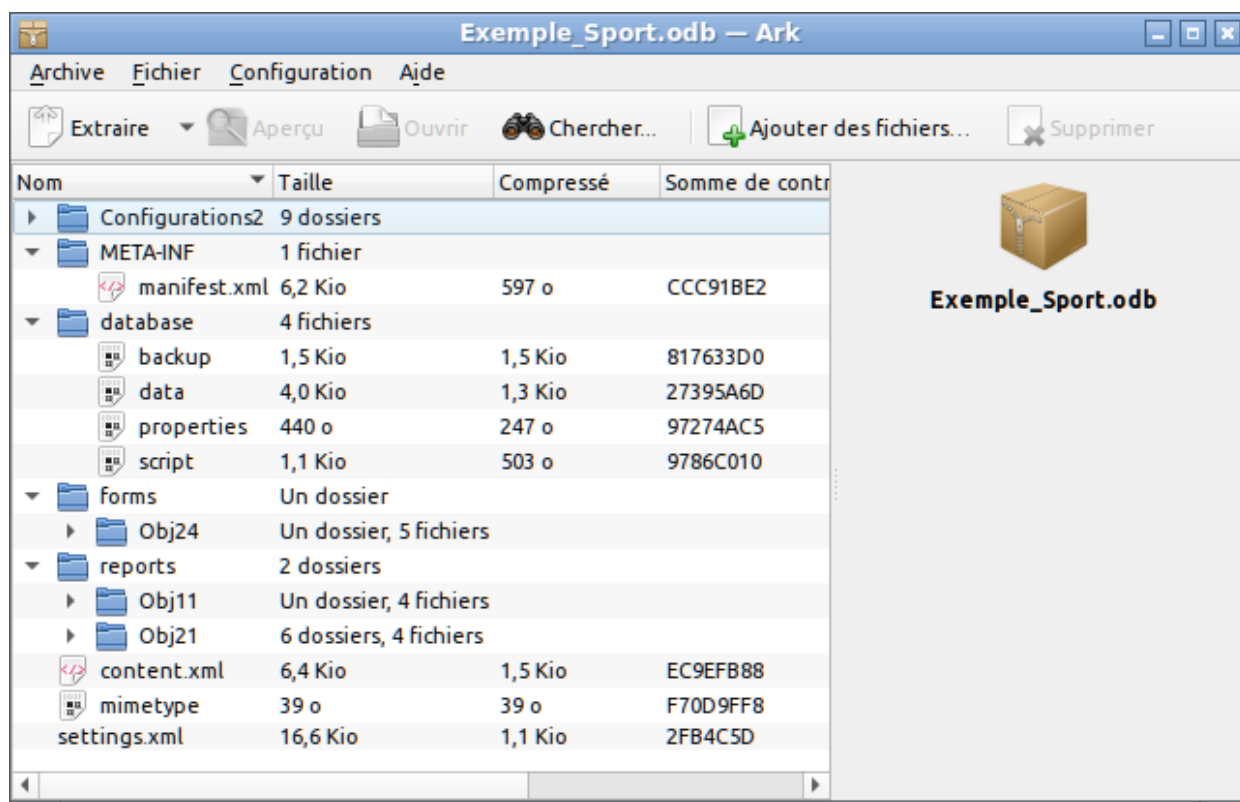
- 9) Ouvrez le fichier de base de données dans LibreOffice. Vous devriez pouvoir à nouveau accéder à vos tables.
- 10) La quantité de vos requêtes, formulaires et rapports pouvant être récupérés de la même manière doit faire l'objet de tests supplémentaires.

Voir aussi : <http://forum.openoffice.org/en/forum/viewtopic.php?f=83&t=17125>

Informations supplémentaires sur les fichiers d'archive de base de données

Dans la pratique, un fichier d'archive de base de données contient non seulement le dossier de base de la base de données et le dossier META-INF qui est spécifié pour le format OpenDocument, mais également des dossiers supplémentaires pour stocker les formulaires et les rapports. Une description de la structure de base du format OpenDocument peut être trouvée sur https://en.wikipedia.org/wiki/OpenDocument_technical_specification.

La vue suivante montre une base de données contenant des tables, des formulaires et des états. Il n'est pas évident que la base de données contienne également une requête. Les requêtes ne sont pas stockées dans des dossiers séparés mais dans le fichier **content.xml**. Les informations nécessaires pour exécuter une requête sont un simple morceau de code SQL.



Fichier de base de données contenant des informations stockées.

Voici un aperçu de l'un des fichiers d'archive de la base de données.

mimetype

```
application/vnd.oasis.opendocument.base
```

Ce petit fichier texte ne contient que l'avertissement que ce fichier d'archive est un fichier de base de données au format OpenDocument.

content.xml pour une base de données sans contenu

```
<?xml version="1.0" encoding="UTF-8"?>
<office: document-content
```

```

xmlns: office="urn: oasis: names: tc: opendocument: xmlns: office:1.0" xmlns:
style="urn: oasis: names: tc: opendocument: xmlns: style:1.0" xmlns: text="urn:
oasis: names: tc: opendocument: xmlns: text:1.0" xmlns: table="urn: oasis: names: tc:
opendocument: xmlns: table:1.0" xmlns: draw="urn: oasis: names: tc: opendocument:
xmlns: drawing:1.0" xmlns: fo="urn: oasis: names: tc: opendocument: xmlns: xsl-fo-
compatible:1.0" xmlns: xlink="http://www.w3.org/1999/xlink"
xmlns: dc="http://purl.org/dc/elements/1.1/" xmlns: meta="urn: oasis: names: tc:
opendocument: xmlns: meta:1.0" xmlns: number="urn: oasis: names: tc: opendocument:
xmlns: datastyle:1.0" xmlns: svg="urn: oasis: names: tc: opendocument: xmlns: svg-
compatible:1.0" xmlns: chart="urn: oasis: names: tc: opendocument: xmlns: chart:1.0"
xmlns: dr3d="urn: oasis: names: tc: opendocument: xmlns: dr3d:1.0" xmlns:
math="http://www.w3.org/1998/Math/MathML"
xmlns: form="urn: oasis: names: tc: opendocument: xmlns: form:1.0" xmlns:
script="urn: oasis: names: tc: opendocument: xmlns: script:1.0"
xmlns: ooo="http://openoffice.org/2004/office"
xmlns: oow="http://openoffice.org/2004/writer"
xmlns: ooc="http://openoffice.org/2004/calc"
xmlns: dom="http://www.w3.org/2001/xml-events"
xmlns: db="urn: oasis: names: tc: opendocument: xmlns: database:1.0" xmlns:
xforms="http://www.w3.org/2002/xforms"
xmlns: xsd="http://www.w3.org/2001/XMLSchema"
xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns: rpt="http://openoffice.org/2005/report"
xmlns: of="urn: oasis: names: tc: opendocument: xmlns: of:1.2" xmlns:
xhtml="http://www.w3.org/1999/xhtml"
xmlns: grddl="http://www.w3.org/2003/g/data-view#"
xmlns: tableooo="http://openoffice.org/2009/table"
xmlns: drawooo="http://openoffice.org/2010/draw"
xmlns: calcext="urn: org: documentfoundation: names: experimental: calc: xmlns:
calcext:1.0" xmlns: field="urn: openoffice: names: experimental: ooo-ms-interop:
xmlns: field:1.0" xmlns: formx="urn: openoffice: names: experimental: ooxml-odf-
interop: xmlns: form:1.0" xmlns: css3t="http://www.w3.org/TR/css3-text/"
office: version="1.2">
  <office: scripts/>
  <office: font-face-decls/>
  <office: automatic-styles/>
  <office: body>
    <office: database>
      <db: data-source>
        <db: connection-data>
          <db: connection-resource xlink: href="sdbc: embedded: hsqldb"/>
          <db: login db: is-password-required="false"/>
        </db: connection-data>
        <db: driver-settings>
          db: system-driver-settings=""
          db: base-dn=""
          db: parameter-name-substitution="false"/>
        <db: application-connection-settings>
          db: is-table-name-length-limited="false"
          db: append-table-alias-name="false"
          db: max-row-count="100">
            <db: table-filter>
              <db: table-include-filter>
                <db: table-filter-pattern>%</db: table-filter-pattern>
              </db: table-include-filter>
            </db: table-filter>
          </db: application-connection-settings>
        </db: data-source>
      </office: database>
    </office: body>
  </office: document-content>

```

Il commence par la version XML et le jeu de caractères utilisé. Tout ce qui suit est en fait une seule ligne continue. La vue préparée ci-dessus devrait rendre les choses plus claires. Les éléments qui vont ensemble sont encadrés par des balises.

Les définitions initiales commençant par xmlns : (espace de noms XML) donnent les espaces de noms accessibles depuis l'intérieur du fichier. Ensuite, des termes un peu plus concrets sont envisagés. Ici, il devient clair que nous avons affaire à une base de données interne HSQLDB, et qu'un mot de passe n'est pas requis pour l'accès.

content.xml pour une base de données avec des contenus

Le contenu suivant n'est qu'un extrait du fichier content.xml, pour clarifier sa structure.

```
<office: scripts/>
<office: font-face-decls>
  <style: font-face style: name="F" svg: font-family=""/>
</office: font-face-decls>
<office: automatic-styles>
  <style: style
    style: name="co1"
    style: family="table-column"
    style: data-style-name="N0"/>
  <style: style
    style: name="co2"
    style: family="table-column"
    style: data-style-name="N107"/>
  <style: style style: name="ce1" style: family="table-cell">
    <style: paragraph-properties fo: text-align="start"/>
  </style: style>
  <number: number-style style: name="N0" number: language="de" number: country="DE">
    <number: number number: min-integer-digits="1"/>
  </number: number-style>
  <number: currency-style
    style: name="N107P0"
    style: volatile="true"
    number: language="de"
    number: country="DE">
    <number: number
      number: decimal-places="2"
      number: min-integer-digits="1"
      number: grouping="true"/>
    <number: text> </number: text>
    <number: currency-symbol
      number: language="de"
      number: country="DE">€
    </number: currency-symbol>
  </number: currency-style>
```

Ici, un champ est défini comme un champ de devise. Le nombre de décimales est indiqué, la séparation entre les nombres et le symbole monétaire et le symbole monétaire lui-même.

```
<number: currency-style
  style: name="N107"
  number: language="de"
  number: country="DE">
  <style: text-properties fo: color="#ff0000"/>
  <number: text>-</number: text>
  <number: number
    number: decimal-places="2"
    number: min-integer-digits="1"
    number: grouping="true"/>
  <number: text> </number: text>
  <number: currency-symbol
    number: language="de"
    number: country="DE">€
  </number: currency-symbol>
  <style: map style: condition="value()>=0" style: apply-style-name="N107P0"/>
</number: currency-style>
```

Le deuxième extrait indique que jusqu'à une valeur particulière, la devise doit apparaître en rouge (« ff0000 »).

```

</office: automatic-styles>
<office: body>
  <office: database>
    <db: data-source>

```

Cette entrée du fichier content.xml ci-dessus, avec toutes ses sous-entrées, correspond à un fichier d'archive de base de données vide.

```

    </db: data-source>
    <db: forms>
      <db: component
        db: name="Receipts"
        xlink: href="forms/Obj12"
        db: as-template="false"/>
    </db: forms>

```

Le fichier d'archive de la base de données contient une sous-section dans laquelle les détails d'un formulaire sont stockés. Le formulaire est désigné dans l'interface utilisateur sous le nom de *Receipts*.

```

    <db: reports>
      <db: component
        db : name="Receipts"
        xlink: href="reports/Obj12"
        db: as-template="false"/>
    </db: reports>

```

Le fichier d'archive de la base de données contient également une sous-section dans laquelle les détails d'un rapport sont stockés. Le rapport est également désigné dans l'interface utilisateur sous le nom de *Receipts*.

```

    <db: queries>
      <db: query
        db: name="Sales_calc"
        db: command="SELECT &quot; a&quot;.*, (SELECT &quot; Price&quot; *
          &quot; a&quot;.&quot; Total&quot; FROM &quot; Stock&quot; WHERE
          &quot; ID&quot; = &quot; a&quot;.&quot; Stock_ID&quot;) AS
          &quot; Total*Price&quot; FROM &quot; Sales&quot; AS &quot; a&quot;"/>
      </db: queries>

```

Toutes les requêtes sont stockées directement dans content.xml. " représente les guillemets doubles. La requête ci-dessus dans cet exemple est en fait assez compliquée avec de nombreuses sous-requêtes corrélées. Il est reproduit ici sous une forme abrégée.

```

    <db: table-representations>
      <db: table-representation db: name="Receipts"/>
      <db: table-representation db: name="Sales"/>
      <db: table-representation db: name="Stock">
        <db: columns>
          <db: column
            db: name="ID"
            db: style-name="co1"
            db: default-cell-style-name="ce1"/>
          <db: column
            db: name="MWSt"
            db: style-name="co1"
            db: default-cell-style-name="ce1"/>
          <db: column
            db: name="Price"
            db: style-name="co2"
            db: default-cell-style-name="ce1"/>
          <db: column
            db: name="Stock"
            db: style-name="co1"
            db: default-cell-style-name="ce1"/>
        </db: columns>
      </db: table-representation>
    </db: table-representations>

```

Cela montre comment les différents tableaux doivent être affichés. Ici, les propriétés d'affichage de colonnes particulières sont stockées : dans cet exemple, les paramètres de la table Stock avec ses champs – ID, MWSt, etc. – sont stockés. Apparemment, quelque chose a été directement entré ici, modifiant un peu les colonnes du tableau.

```
</office: database>
</office: body>
```

Fondamentalement, content.xml stocke directement le contenu des requêtes et des informations sur l'apparence visuelle des tables. En outre, il existe une définition de la connexion à la base de données. Viennent enfin les informations sur les formulaires et les rapports.

settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<office: document-settings
xmlns: office="urn: oasis: names: tc: opendocument: xmlns: office:1.0" xmlns:
table="urn: oasis: names: tc: opendocument: xmlns: table:1.0" xmlns:
xlink="http://www.w3.org/1999/xlink" xmlns: number="urn: oasis: names: tc:
opendocument: xmlns: datastyle:1.0" xmlns: svg="http://www.w3.org/2000/svg" xmlns:
config="urn: oasis: names: tc: opendocument: xmlns: config:1.0"
xmlns: ooo="http://openoffice.org/2004/office" xmlns: db="urn: oasis: names: tc:
opendocument: xmlns: database:1.0"
office: version="1.2"/>
```

Pour une base de données sans autre contenu, seules les définitions de base sont stockées ici. Avec le contenu, divers paramètres sont également enregistrés. Après le début de la définition ci-dessus, les paramètres suivants de la base de données d'exemple sont stockés.

```
<office: settings>
  <config: config-item-set config: name="ooo: view-settings">
    <config: config-item-set config: name="Queries">
      <config: config-item-set config: name="Calculate_sales">
        <config: config-item-set config: name="Tables">
          <config: config-item-set config: name="Table1">
            <config: config-item config: name="WindowName"
              config: type="string">Verkauf</config: config-item>
            <config: config-item config: name="WindowLeft"
              config: type="int">153</config: config-item>
            <config: config-item config: name="ShowAll"
              config: type="boolean">true</config: config-item>
            <config: config-item config: name="WindowTop"
              config: type="int">17</config: config-item>
            <config: config-item config: name="WindowWidth"
              config: type="int">120</config: config-item>
            <config: config-item config: name="WindowHeight"
              config: type="int">120</config: config-item>
            <config: config-item config: name="ComposedName"
              config: type="string">Verkauf</config: config-item>
            <config: config-item config: name="TableName"
              config: type="string">Verkauf</config: config-item>
          </config: config-item-set>
        </config: config-item-set>
      <config: config-item config: name="SplitterPosition"
        config: type="int">105</config: config-item>
      <config: config-item config: name="VisibleRows"
        config: type="int">1024</config: config-item>
    </config: config-item-set>
  </config: config-item-set>
  <config: config-item-set config: name="ooo: configuration-settings">
    <config: config-item-set config: name="layout-settings">
      <config: config-item-set config: name="Tables">
        <config: config-item-set config: name="Table1">
          <config: config-item config: name="WindowName"
            config: type="string">Verkauf</config: config-item>
          <config: config-item config: name="WindowLeft"
            config: type="int">186</config: config-item>
          <config: config-item config: name="ShowAll"
            config: type="boolean">true</config: config-item>
        </config: config-item-set>
      </config: config-item-set>
    </config: config-item-set>
  </config: config-item-set>
</office: settings>
```

```

        config: type="boolean">false</config: config-item>
    <config: config-item config: name="WindowTop"
        config: type="int">17</config: config-item>
    <config: config-item config: name="WindowWidth"
        config: type="int">120</config: config-item>
    <config: config-item config: name="WindowHeight"
        config: type="int">120</config: config-item>
    <config: config-item config: name="ComposedName"
        config: type="string">Verkauf</config: config-item>
    <config: config-item config: name="TableName"
        config: type="string">Sales</config: config-item>
</config: config-item-set>
<config: config-item-set config: name="Table2">
    ... (identical config: type-Points as "Table1"
    <config: config-item config: name="TableName"
        config: type="string">Ware</config: config-item>
</config: config-item-set>
<config: config-item-set config: name="Table3">
    ... (identical config: type-Points as "Table1"
    <config: config-item config: name="TableName"
        config: type="string">Receipts</config: config-item>
</config: config-item-set>
</config: config-item-set>
</config: config-item-set>
</office: settings>

```

L'ensemble de la vue d'ensemble concerne différentes vues des fenêtres pour la requête Calculate_sales et les tables Sales, Stock et Receipts. Les deux derniers sont présentés ici sous une forme abrégée. Si ces paramètres étaient absents dans un fichier *.odf défectueux, cela n'aurait pas d'importance. Ils seraient recréés lors de la prochaine ouverture de la fenêtre correspondante.

META-INF/manifest.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<manifest: manifest xmlns: manifest="urn: oasis: names: tc: opendocument: xmlns:
manifest:1.0">
    <manifest: file-entry
        manifest: full-path="/"
        manifest: media-type="application/vnd.oasis.opendocument.base"/>
    <manifest: file-entry
        manifest: full-path="database/script"
        manifest: media-type=""/>
    <manifest: file-entry
        manifest: full-path="database/properties"
        manifest: media-type=""/>
    <manifest: file-entry
        manifest: full-path="settings.xml"
        manifest: media-type="text/xml"/>
    <manifest: file-entry
        manifest: full-path="content.xml"
        manifest: media-type="text/xml"/>
</manifest: manifest>

```

Ce fichier dans le dossier META-INF donne le dossier de contenu pour toute l'archive de la base de données. Comme ce fichier traite d'une base de données vide, il n'y a que cinq entrées de fichier. Une archive de base de données qui contient des formulaires et des rapports aura un fichier META-INF beaucoup plus compliqué.

database/properties

```

#HSQL Database Engine 1.8.0.10
#Sun Jul 14 18:02:08 CEST 2013
hsqldb.script_format=0

```



```
runtime.gc_interval=0
sql.enforce_strict_size=true
hsqldb.cache_size_scale=8
readonly=false
hsqldb.nio_data_file=false
hsqldb.cache_scale=13
version=1.8.0
hsqldb.default_table_type=cached
hsqldb.cache_file_scale=1
hsqldb.lock_file=true
hsqldb.log_size=10
modified=no
hsqldb.cache_version=1.7.0
hsqldb.original_version=1.8.0
hsqldb.compatible_version=1.8.0
```

Le fichier de propriétés contient les paramètres de base de la base de données interne HSQLDB.

database/script

```
SET DATABASE COLLATION "French"
CREATE SCHEMA PUBLIC AUTHORIZATION DBA
CREATE USER SA PASSWORD ""
GRANT DBA TO SA
SET WRITE_DELAY 60
```

Le fichier de script contient les paramètres par défaut pour la connexion à la base de données, le paramètre de langue, etc. L'utilisateur SA, qui sera décrit plus loin, apparaît ici.

Dans une base de données avec contenu, ce fichier contient les définitions de table de base.

```
SET DATABASE COLLATION "French"
CREATE SCHEMA PUBLIC AUTHORIZATION DBA
```

Les tables sont définies avant que l'utilisateur de la base de données ne soit défini. Les tables sont d'abord créées dans le cache avec leurs champs.

```
CREATE CACHED TABLE "Stock"
("ID" INTEGER GENERATED BY DEFAULT AS IDENTITY(START WITH 0) NOT NULL
PRIMARY KEY, "Stock" VARCHAR(50), "Price" DECIMAL(8, 2), "MwSt" TINYINT)
CREATE CACHED TABLE "Sales"
("ID" INTEGER GENERATED BY DEFAULT AS IDENTITY(START WITH 0) NOT NULL
PRIMARY KEY, "Total" TINYINT, "Stock_ID" INTEGER, "Receipt_ID" INTEGER,
CONSTRAINT SYS_FK_59 FOREIGN KEY("Stock_ID") REFERENCES "Stock"("ID"))
CREATE CACHED TABLE "Receipts"
("ID" INTEGER GENERATED BY DEFAULT AS IDENTITY(START WITH 0) NOT NULL
PRIMARY KEY, "Date" DATE)
```

Des modifications sont ensuite apportées à la table pour garantir la cohérence des relations (RÉFÉRENCES).

```
ALTER TABLE "Sales" ADD CONSTRAINT SYS_FK_76 FOREIGN KEY("Receipt_ID")
REFERENCES "Receipts"("ID")
SET TABLE "Stock" INDEX'608 20'
SET TABLE "Sales" INDEX'1872 1656 1872 12'
SET TABLE "Receipts" INDEX'2232 1'
```

Après avoir défini la position de l'index dans le fichier de données (il n'apparaît ici que dans le fichier de script mais n'est jamais réellement entré directement dans SQL), les champs incrémentés automatiquement dans les tables (AutoValues) sont configurés de manière à fournir le prochain valeur à l'entrée d'un nouvel enregistrement. Supposons que la dernière valeur entrée dans le champ ID de la table Stock soit 19. L'auto-incrémentation commence alors à 20.

```
ALTER TABLE "Stock" ALTER COLUMN "ID" RESTART WITH 20
ALTER TABLE "Sales" ALTER COLUMN "ID" RESTART WITH 12
```

```
ALTER TABLE "Receipts" ALTER COLUMN "ID" RESTART WITH 1
CREATE USER SA PASSWORD ""
GRANT DBA TO SA
SET WRITE_DELAY 60
```

Gestion de la base de données interne Firebird

La base de données interne Firebird n'est pour le moment disponible qu'en tant que fonction expérimentale. Pour créer une telle base de données ou pour en modifier une qui a été créée, vous devez sélectionner **Outils> Options> LibreOffice> Avancé> Fonctionnalités optionnelles> Activer les fonctionnalités expérimentales (peut-être instables)**. Ce chemin illustre bien qu'une telle base de données n'est pas adaptée à un usage quotidien.

Le lien suivant permet de signaler et de traiter des bogues importants dans la base de données interne Firebird avec l'équipe LibreOffice : [Reporting bugs for Firebird in Base](#).

Les utilisateurs remarqueront les différences suivantes par rapport à HSQLDB :

- 1) Si un champ est de type Integer puis déclaré comme clé primaire, il semble possible de lui donner une valeur auto-incrémentée. Cependant, lors de l'enregistrement, ce paramètre disparaît sans préavis.
- 2) Lorsque de nouveaux enregistrements sont saisis, ils ne sont pas automatiquement enregistrés dans la base de données. Le bouton Enregistrer doit être utilisé pour chaque entrée. Dans HSQLDB intégré, la sauvegarde explicite des enregistrements n'est pas nécessaire.
- 3) Les alias sont complètement ignorés dans les requêtes. Un alias peut être créé mais il n'apparaîtra pas dans l'en-tête de table de la requête.
- 4) Il n'est pas possible de créer des conditions, bien que les bases de données externes Firebird les prennent en charge.
- 5) Les types de données décimal et numérique sont actuellement défectueux. Ce sont les seuls types qui garantissent des valeurs précises, en particulier lorsqu'il y a des décimales. Ce sont donc les champs préférés pour les valeurs monétaires. À l'heure actuelle, seules les valeurs comportant au plus une décimale peuvent être saisies.

Rendre AutoChamp disponible

Le code suivant, entré en utilisant **Outils> SQL**, peut aider à résoudre le problème des valeurs automatiques non fournies.

```
CREATE TABLE "Table1" ("ID" INTEGER NOT NULL PRIMARY KEY, "Nom" VARCHAR(20)
NOT NULL) ;
CREATE GENERATOR GEN_T1_ID ;
SET GENERATOR GEN_T1_ID TO 0 ;
```

Après cela, la fenêtre de saisie SQL doit être fermée et **Affichage> Actualiser les tables** sélectionné. Le déclencheur suivant ne peut être créé que lorsque la table apparaît, et dans certains cas seulement après une tentative (infructueuse) de création d'une entrée.

```
CREATE TRIGGER T1_BI FOR "Table1" ACTIVE BEFORE INSERT POSITION 0 AS
BEGIN
IF (NEW. ID IS NULL) THEN NEW. ID = GEN_ID(GEN_T1_ID, 1) ;
END ;
```

Même après cela, de nombreuses entrées dans Noms peuvent être effectuées dans la table sans créer d'entrée dans ID. Le champ ID indique simplement 0. Les valeurs affectées réelles ne sont affichées que lorsque vous appuyez sur **Actualiser**. Le déclencheur fournit des valeurs commençant par 1.