LibreOffice
The Document Foundation

Base

# *Kapitel 2*
# *Datenbank erstellen*

# Copyright

Dieses Dokument unterliegt dem Copyright © 2014. Die Beitragenden sind unten aufgeführt. Sie dürfen dieses Dokument unter den Bedingungen der GNU General Public License (*http://www.-gnu.org/licenses/gpl.html*), Version 3 oder höher, oder der Creative Commons Attribution License (*http://creativecommons.org/licenses/by/3.0/*), Version 3.0 oder höher, verändern und/oder weitergeben.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.

Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das Symbol (R) in diesem Buch nicht verwendet.

## Mitwirkende/Autoren

Robert Großkopf Jost Lange Michael Niedermair

Jochen Schiffers

## Rückmeldung (Feedback)

Kommentare oder Vorschläge zu diesem Dokument können Sie in deutscher Sprache an die Adresse *discuss@de.libreoffice.org* senden.

| **Vorsicht** | Alles, was an eine Mailingliste geschickt wird, inklusive der E-Mail-Adresse und anderer persönlicher Daten, die die E-Mail enthält, wird öffentlich archiviert und kann nicht gelöscht werden. Also, schreiben Sie mit Bedacht! |
|---|---|

## Datum der Veröffentlichung und Softwareversion

Veröffentlicht am 15.8.2014. Basierend auf der LibreOffice Version 4.3.

## Anmerkung für Macintosh Nutzer

Einige Tastenbelegungen (Tastenkürzel) und Menüeinträge unterscheiden sich zwischen der Macintosh Version und denen für Windows- und Linux-Rechnern. Die unten stehende Tabelle gibt Ihnen einige grundlegende Hinweise dazu. Eine ausführlichere Aufstellung dazu finden Sie in der Hilfedatei des jeweiligen Moduls.

| ***Windows/Linux*** | ***entspricht am Mac*** | ***Effekt*** |
|---|---|---|
| Menü-Auswahl Extras → Optionen | **LibreOffice → Einstellungen** | Zugriff auf die Programmoptionen |
| Rechts-Klick | *Control*+Klick | Öffnen eines Kontextmenüs |
| *Ctrl* (Control) oder *Strg* (Steuerung) | ⌘ (Command) | Tastenkürzel in Verbindung mit anderen Tasten |
| *F5* | *Shift*+⌘+*F5* | Öffnen des Dokumentnavigator-Dialogs |
| *F11* | ⌘+*T* | Öffnen des Formatvorlagen-Dialogs |

# Inhalt


Allgemeines bezüglich der Erstellung einer Datenbank ..... 4

Neue Datenbank als interne Datenbank ..... 4

Zugriff auf externe Datenbanken ..... 6

- MySQL/MariaDB-Datenbanken ..... 7
  - Erstellen eines Nutzers und einer Datenbank ..... 7
  - MySQL-Verbindung direkt mit der Extension ..... 8
  - MySQL-Verbindung über JDBC ..... 8
  - MySQL-Verbindung über ODBC ..... 8
  - Verbindung zur MySQL-Datenbank über den Datenbank-Assistenten ..... 9
- PostgreSQL ..... 16
  - Erstellen eines Nutzers und einer Datenbank ..... 16
  - Direkte Verbindung zu Base ..... 17
- dBase-Datenbanken ..... 18
- Tabellendokumente ..... 20
- Thunderbird Adressbuch ..... 21
- Texttabellen ..... 22
- Firebird ..... 24
  - Erstellen eines Nutzers und einer Datenbank ..... 25
  - Firebird-Verbindung über JDBC ..... 25
  - Firebird-Verbindung über ODBC ..... 26

Nachträgliche Bearbeitung der Verbindungseigenschaften ..... 26

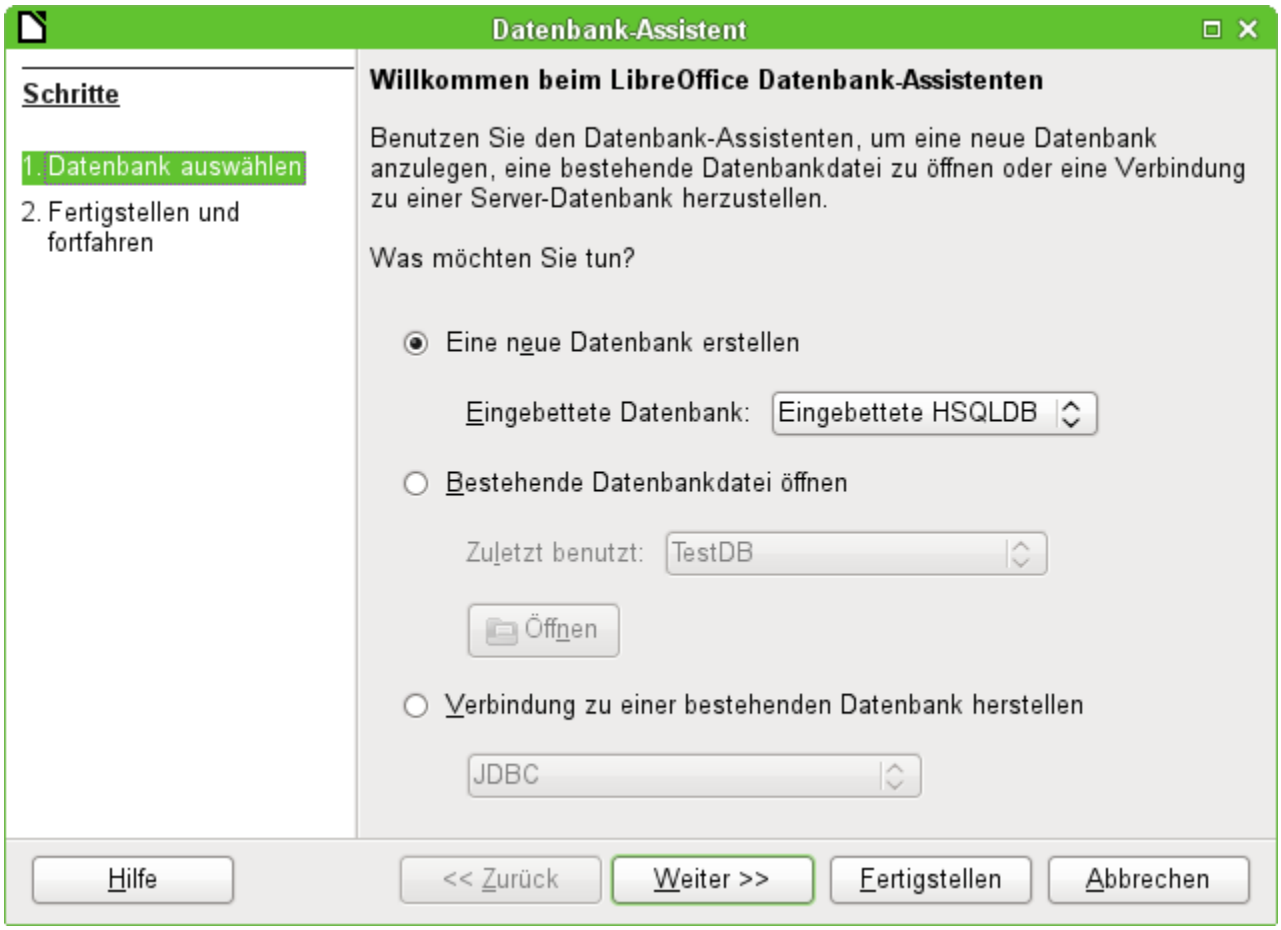
## Allgemeines bezüglich der Erstellung einer Datenbank

In LibreOffice gibt es das Programmmodul «Base». Dies stellt eine grafische Benutzeroberfläche für Datenbanken zur Verfügung. Daneben existiert, in LibreOffice eingebunden, die interne Datenbank «HSQLDB». Diese interne Version der HSQLDB-Datenbank kann nur als Datenbank von einem Nutzer betrieben werden. Die gesamten Daten sind in der *.odb-Datei mit abgespeichert und diese Datei ist nur für einen Benutzer nicht schreibgeschützt zu öffnen.

## Neue Datenbank als interne Datenbank

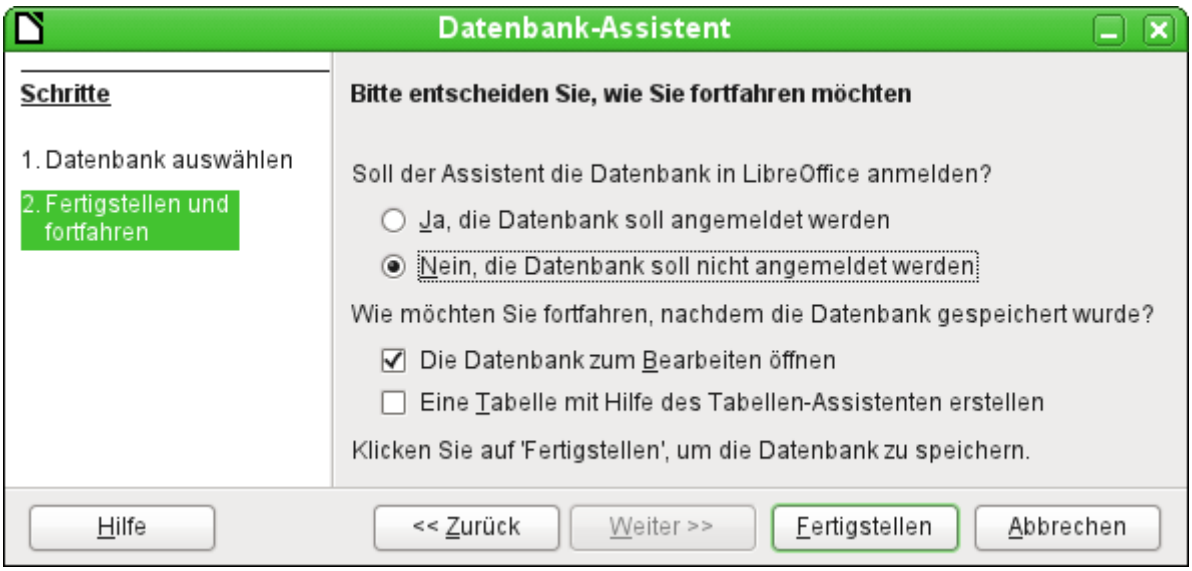
Sofern nicht von vornherein eine Multiuserdatenbank geplant ist oder erst einmal erste Erfahrungen mit einer Datenbank gesammelt werden sollen, ist die interne Datenbank gut nutzbar. Diese lässt sich später auch mit ein paar Kniffen noch zur externen HSQLDB umwandeln, auf die dann auch mehrere User gleichzeitig zugreifen können, wenn der HSQLDB-Server läuft. Eine Beschreibung dazu erfolgt im Anhang dieses Handbuches im Kapitel «Datenbankverbindung zu einer externen HSQLDB».

Die Erstellung einer internen Datenbank erfolgt direkt über den Eingangsbildschirm von LO mit einem Klick auf den Button Datenbank oder über **Datei → Neu → Datenbank.** Der Datenbank-Assistent startet mit dem folgenden Bildschirm:




Das Optionsfeld «Neue Datenbank erstellen» ist für eine neue interne Datenbank angewählt. Standardmäßig ist dies die eingebettete HSQLDB. Ab LO 4.2 steht zusätzlich die interne Firebird-Datenbank zur Verfügung. Diese muss allerdings in LO 4.2 und 4.3 extra aktiviert werden.

Die weiteren Optionen dienen dazu, eine bestehende Datei zu öffnen oder eine Verbindung zu einer externen Datenbank wie z.B. einem Adressbuch, einer MySQL-Datenbank o.ä. aufzubauen.



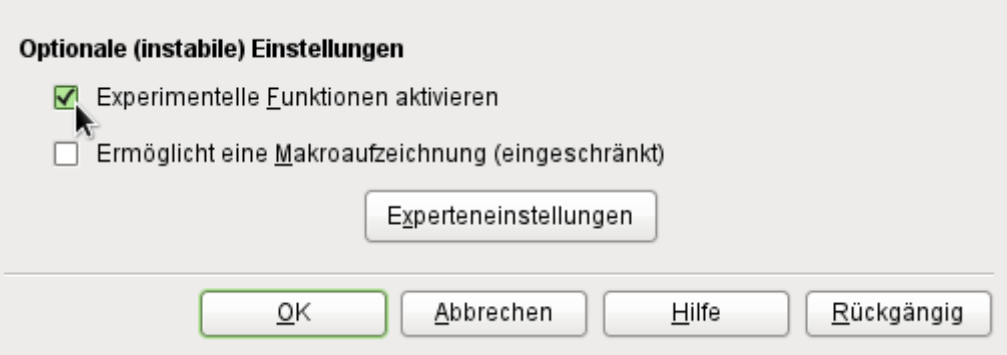

Eine in LibreOffice angemeldete Datenbank kann von den anderen Programmteilen von LibreOffice als Datenquelle genutzt werden (z. B. Serienbrief). Diese Anmeldung kann aber auch zu einem späteren Zeitraum erfolgen. Deshalb ist an dieser Stelle erst einmal das Optionsfeld «Nein, die Datenbank soll nicht angemeldet werden", ausgewählt.

Bei den Markierfeldern ist lediglich «Die Datenbank zum Bearbeiten öffnen» markiert. Assistenten zur Erstellung von Tabellen, Abfragen usw. werden in diesem Handbuch nicht genutzt. Ihre Funktion ist in dem Kapitel aus dem «Erste Schritte Handbuch» «*Einführung in Base*» erklärt.

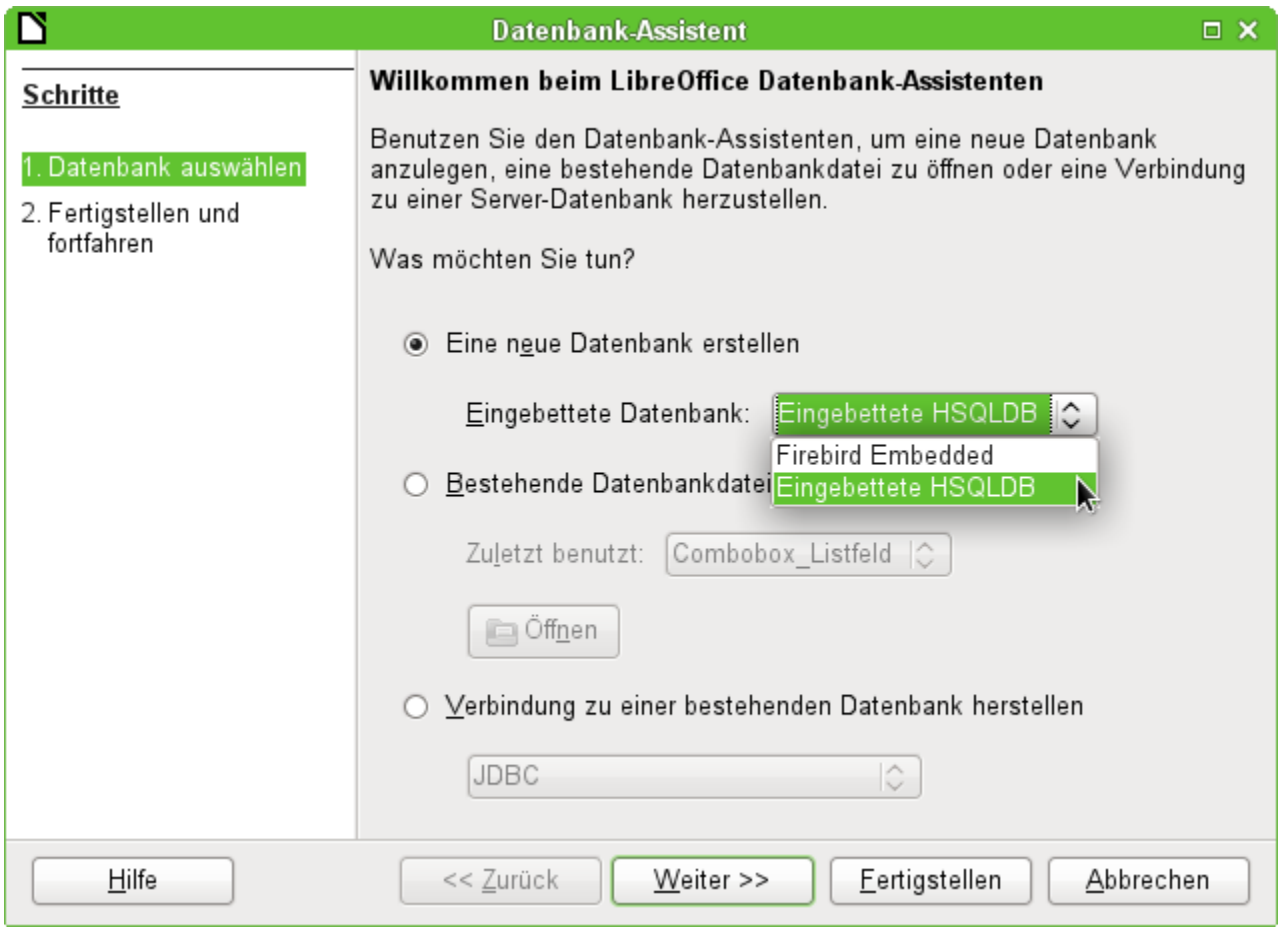
Die Datenbank wird fertiggestellt, indem direkt noch einem Namen und einem Speicherort gefragt wird. An der entsprechenden Astelle wird dann die *.odb-Datei erzeugt, die zur Aufnahme von Daten aus der internen Datenbank, zur Speicherung von Abfragen, Formularen und Berichten gedacht ist.

Im Gegensatz zu den anderen Programmteilen von LibreOffice wird also die Datei abgespeichert, bevor der Nutzer überhaupt irgendwelche für ihn sichtbare Eingaben getätigt hat.

Mit der Version LibreOffice 4.2 wurde zusätzlich zu der internen HSQLDB eine interne Firebird-Datenbank aufgenommen. Um diese bisher noch nicht komplett integrierte Datenbank zu nutzen müssen über **Extras → Optionen → LibreOffice → Erweitert → Optionale (instabile) Einstellungen** die experimentellen Funktionen aktiviert werden:




Danach erscheint in dem Datenbankassistenten neben der internen HSQLDB auch die interne Firebird-Datenbank zur Auswahl:




Die eingebettete Firebird-Datenbank ist allerdings in der Version 4.2 noch weit entfernt von ihren tatsächlichen Möglichkeiten. SQL-Erweiterungen zu der HSQLDB sind noch nicht implementiert; auch die Erstellung automatisch hoch zählender Primärschlüssel klappt noch nicht.

Die HSQLDB ist auch in LibreOffice 4.2 und 4.3 die Standarddatenbank für jeden Nutzer; die Firebird-Datenbank steht aber bereits für erste Experimente bereit. Das Handbuch bezieht sich aus diesem Grunde vorerst weiter auf die interne HSQLDB. Für die interne Firebird-Datenbank ist zusätzlich ein Vergleich der Funktionen und Datentypen mit denen der internen HSQLDB zu dem Paket der Datenbank-Beispiele hinzugefügt worden.

## Zugriff auf externe Datenbanken

Alle externen Datenbanken müssen zuerst einmal existieren. Angenommen es soll der Zugriff auf die Datenbank einer eigenen Homepage erfolgen. Dort muss zuerst einmal die Datenbank selbst mit einem Namen und Zugangsdaten gegründet worden sein, bevor externe Programme darauf zugreifen können.

Existiert nun diese externe Datenbank, so kann sie, je nach vorhandener Treibersoftware, zur Erstellung von Tabellen und anschließender Dateneingabe und -abfrage genutzt werden.

Über **Datei → Neu → Datenbank** wird der Datenbankassistent geöffnet und die Verbindung zu einer bestehenden Datenbank hergestellt. Die Liste der hier aufgeführten Treiber, die mit Datenbanken verbinden, variiert etwas je nach Betriebssystem und Benutzeroberfläche. Immer dabei sind aber

- dBase
- JDBC

- MySQL
- ODBC
- Oracle JDBC
- PostgreSQL
- Tabellendokument
- Text
- sowie verschiedene Adressbücher

Je nach gewählter Datenbank variieren nun die Verbindungseinstellungen. Diese können auch gegebenenfalls später noch korrigiert werden, wenn erst einmal die *.odb-Datei erstellt worden ist.

Bei manchen Datenbanktypen können keine neuen Daten eingegeben werden. Sie sind deshalb nur zum Suchen von Daten geeignet (z.B. Tabellendokument, Adressbücher). Die Beschreibungen in den folgenden Kapiteln beziehen sich ausschließlich auf die Verwendung von LibreOffice Base mit der internen Datenbank HSQLDB. Die meisten Ausführungen lassen sich auf Datenbanken wie MySQL, PostgreSQL etc. übertragen und entsprechend anwenden.

Hier soll nur kurz an ein paar Beispielen vorgestellt werden, wie der Kontakt zu anderen externen Datenbanken hergestellt wird.

## MySQL/MariaDB-Datenbanken

MySQL-Datenbanken oder auch MariaDB-Datenbanken können auf drei verschiedene Weisen mit Base verbunden werden. Die einfachste und schnellste Art ist die direkte Verbindung mit dem MySQL-Connector. Daneben steht noch die Verbindung über JDBC und ODBC zur Verfügung.

### *Erstellen eines Nutzers und einer Datenbank*

Nachdem MySQL bzw. MariaDB installiert ist sollten nacheinander die folgenden Schritte vollzogen werden:

- Der Administartionsuser in MySQL heißt «root». Für Linuxnutzer ist hier wichtig, dass es sich dabei nicht um den Administrator des Linuxsystems handelt. Diesem Nutzer wird direkt nach der Installation erst einmal ein Passwort zugewiesen.
  **`mysql -u root -p`**
  Am Anfang ist kein Passwort vorhanden, so dass bei der Eingabeaufforderung auch nur *Enter* gedrückt wird.
  Es erscheint die Eingabeaufforderung
  **`mysql`**>
  Alle folgenden Eingaben werden jetzt auf dieser mysql-Konsole erstellt. Die Passwörter können sich unterscheiden, je nachdem, ob die Anmeldung von dem momentanen Rechner («localhost») oder von einem anderen Rechner zum MySQL-Serverrechner («host») erfolgt.
  ```
  SET PASSWORD FOR root@localhost=PASSWORD('Passwort');
  SET PASSWORD FOR root@host=PASSWORD('Passwort');
  ```
- Als Sicherheitsmaßnahme werden alle eventuell bestehenden anonymen Nutzer gelöscht.
  ```
  DELETE FROM mysql.user WHERE User='';
  DELETE FROM mysql.db WHERE User='';
  FLUSH PRIVILEGES;
  ```
- Eine Datenbank mit dem Namen «libretest» wird erstellt.
  ```
  CREATE DATABASE libretest;
  ```
- Alle Rechte an der Datenbank «libretest» werden dem Nutzer «lotest» übergeben, der sich durch das Passwort «libre» ausweisen soll.
  ```
  GRANT ALL ON libretest.* TO lotest IDENTIFIED BY 'libre';
  ```

Damit ist die Datenbank vorhanden, mit der die folgenden Verbindungen aufgebaut werden.

### MySQL-Verbindung direkt mit der Extension

Für die **MySQL-Datenbank** existieren Erweiterungen zur direkten Verbindung mit Base. Diese Erweiterungen tragen den Namen **MySQL-Connector**.

- Die mit der 3.3.4 funktionierende Erweiterung («Extension») ist unter *http://extensions.services.openoffice.org/en/project/mysql_connector* (Stand: 1/11) zu finden.
- Ab der Version 3.5 gibt es eine für Mac und Ubuntu-Linux getestete neue Version unter: *http://extensions.libreoffice.org/extension-center/mysql-native-connector-for-mac-osx*
- Eine auch unter anderen Systemen lauffähige Erweiterung für Version nach 3.3.4 steht unter *http://extensions.openoffice.org/en/project/aoo-my-sdbc* zum Download.
- Für Version LO 4.0 ist eine Extension für Windows über *https://dl.dropboxusercontent.com/u/193133/mysql-connector-ooo.oxt* downloadbar.
- Unter folgendem Link sind Erweiterungen verfügbar, die zwar für AOO vorgesehen sind, aber zumindest unter Linux einwandfrei mit LO 4.0.* zusammenarbeiten: *http://code.google.com/a/apache-extras.org/p/aoo-my-sdbc/* .

| **Vorsicht** | Ein Kontakt zu LO 4.1.* kommt zwar mit dem AOO-Connector zustande, aber da sich ab der Version LO 4.1 die Beschreibung der Sekundenbruchteile von LO und AOO unterscheiden, werden falsche Werte im Bereich der Zeiten und auch von kompletten Timestamp-Feldern angezeigt.<br>Verwenden sie also bitte immer die Version, die lt. Beschreibung auch zur LO-Version passt. Achten sie besonders beim Wechsel auf LO 4.1 darauf, ob überhaupt ein entsprechender Connector zur Verfügung steht. |
|---|---|

### MySQL-Verbindung über JDBC

Als allgemeiner Zugang zu MySQL ist der Zugang über JDBC oder ODBC zu wählen. Um den JDBC-Zugang nutzen zu können, wird der mysql-connector-java.jar in der jeweils zur Datenbank passenden Fassung benötigt. Dieses Java-Archiv wird am besten in das Verzeichnis kopiert, aus dem auch die aktuelle Java-Version von LibreOffice geladen wird. Als Speicherort bietet sich das Unterverzeichnis «...Javapfad.../lib/ext/» an. In Linux-Systemen wird dies durch die Softwareverwaltung oft automatisch erledigt.

Gegebenenfalls kann das Java-Archiv auch separat über **Extras → Optionen → LibreOffice → Erweitert → Java Optionen→ Class-Pfad** in den Class-Path aus jeder beliebigen Stelle der Festplatte übernommen werden. Für diese Option sind dann auch keine Systemverwalterrechte notwendig.

### MySQL-Verbindung über ODBC

Für eine Verbindung über ODBC muss natürlich erst einmal die entsprechende ODBC-Software installiert sein. Details dazu werden hier nicht weiter beschrieben.

Nach Installation der entsprechenden Software kann es passieren, dass LO den Dienst verweigert, weil es die libodbc.so nicht findet. Hier existiert in den meisten Systemen inzwischen die libodbc.so.2 . Auf diese Datei muss ein entsprechender Verweis mit dem Namen libodbc.so in dem gleichen Verzeichnis abgespeichert werden.

In den für das System notwendigen Dateien odbcinst.ini und odbc.ini müssen Einträge ähnlich den folgenden existieren:

**odbinst.ini**

```
[MySQL]
Description = ODBC Driver for MySQL
Driver = /usr/lib/libmyodbc5.so
```

**odbc.ini**

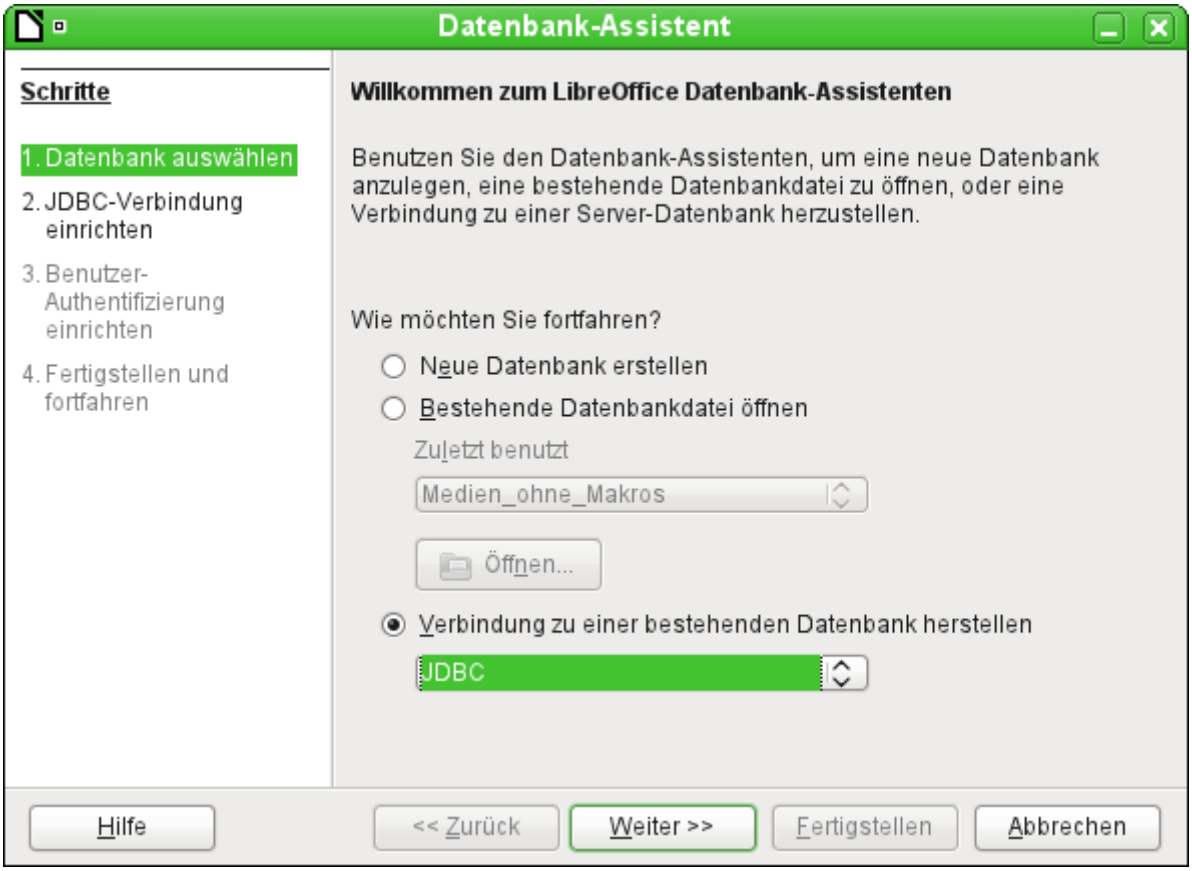
```
[MySQL-test]
Description = MySQL database test
Driver = MySQL
Server = localhost
Database = libretest
Port = 3306
Socket =
Option = 3
Charset = UTF8
```

Diese beiden Dateien liegen bei einem Linux-System im Verzeichnis /etc/unixODBC. Ohne den Eintrag zur Art des verwendeten Zeichensatzes kommt es bei Umlauten zu Problemen, auch wenn die Einstellungen in MySQL/MariaDB und Base übereinstimmen.

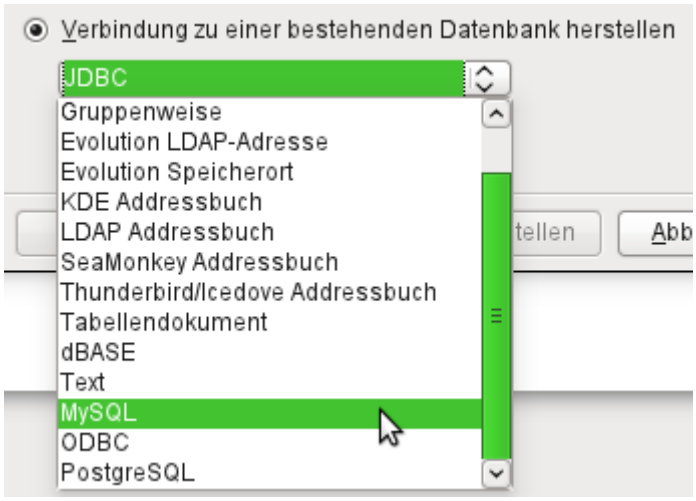
Details zu den Verbindungsparametern sind im *MySQL-Referenzhandbuch* zu finden.

### *Verbindung zur MySQL-Datenbank über den Datenbank-Assistenten*

Der Zugriff auf eine existierende MySQL-Datenbank erfolgt mit der direkten Verbindung in den folgenden Schritten:

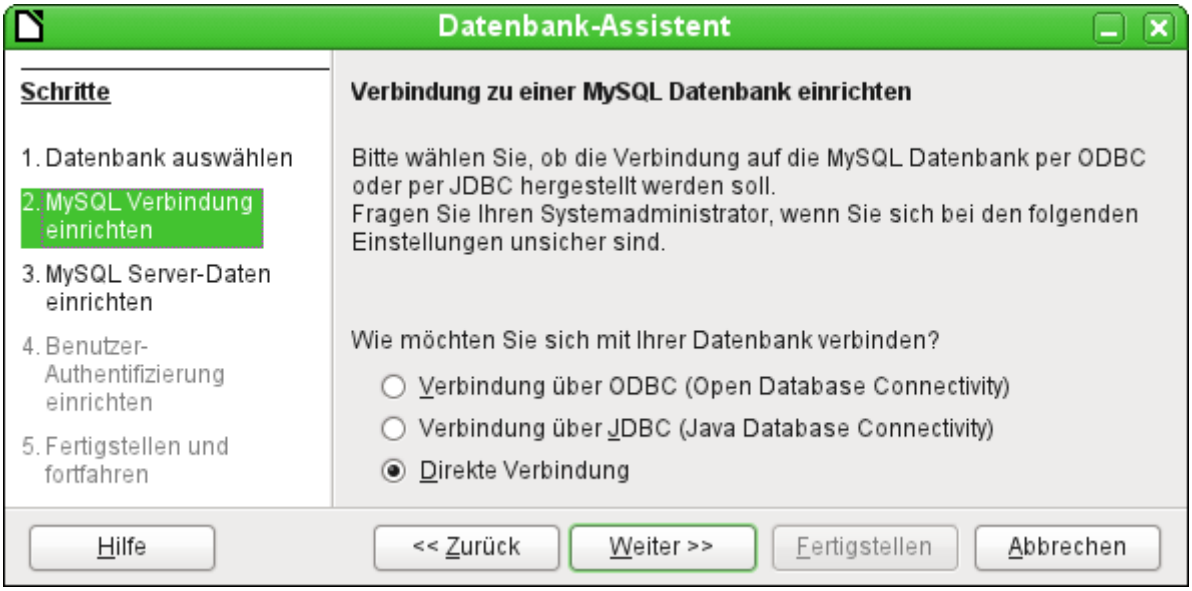


Über «*Neue Datenbank erstellen*» ist nur die Erstellung einer Datenbank im internen HSQLDB-Format möglich. Die Zusammenarbeit mit anderen Datenbanken kann nur erfolgen, wenn die Datenbank selbst bereits existiert. Dazu muss also «*Verbindung zu einer bestehenden Datenbank herstellen*» gewählt werden.




Hier wird aus den, teilweise betriebssystemspezifischen, Datenbanken die MySQL-Variante ausgewählt.

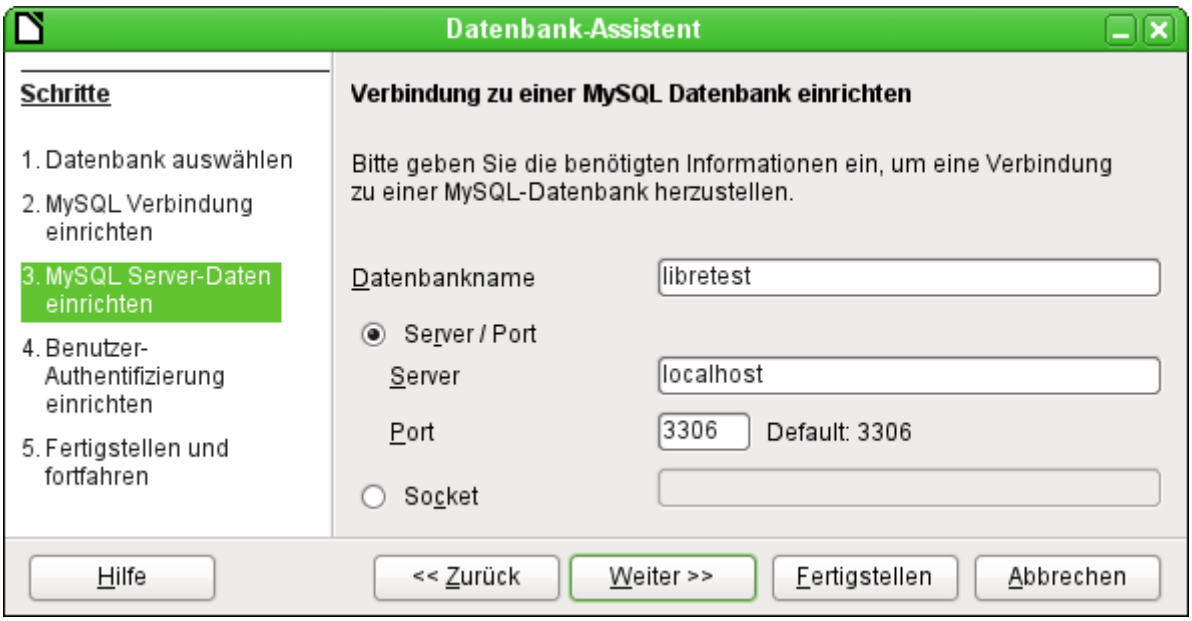
## Die direkte Verbindung




Der Kontakt zu MySQL kann über ODBC oder JDBC hergestellt werden, solange kein nativer[1] MySQL-Connector als direkte Verbindung installiert ist oder unterstützt wird. Ist die Erweiterung für eine direkte Verbindung installiert, so wird «Direkte Verbindung» standardmäßig voreingestellt.

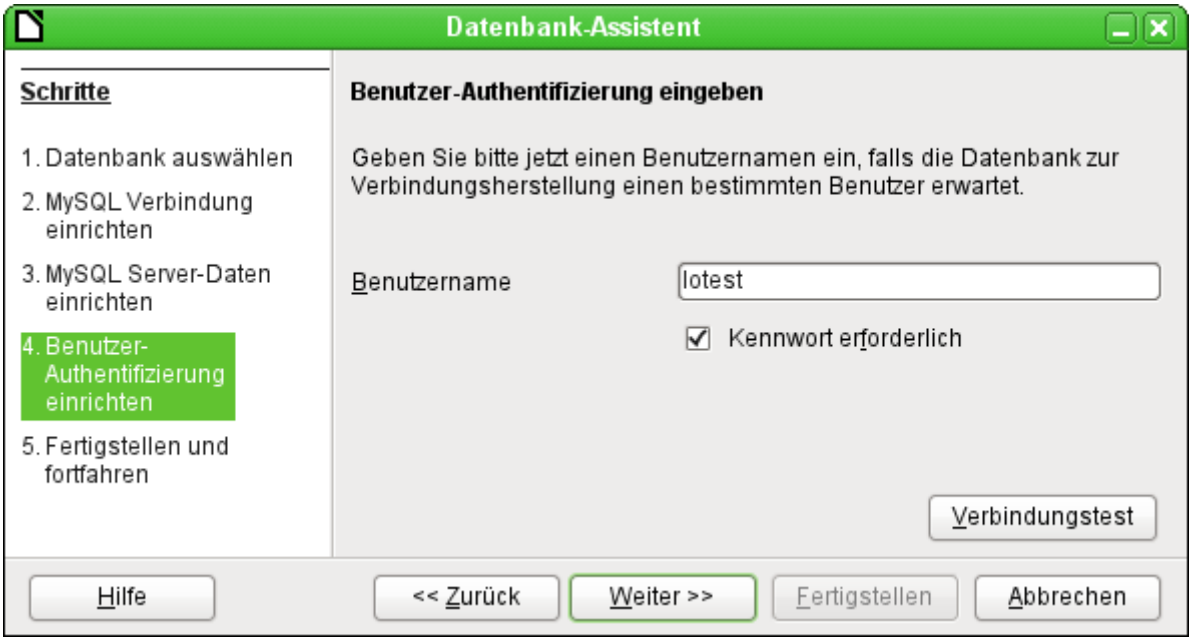
Die direkte Verbindung ist diejenige, die von der Geschwindigkeit und von der Funktionalität her am besten gewählt werden sollte.

1 «nativ» – unverändert, unmodifiziert

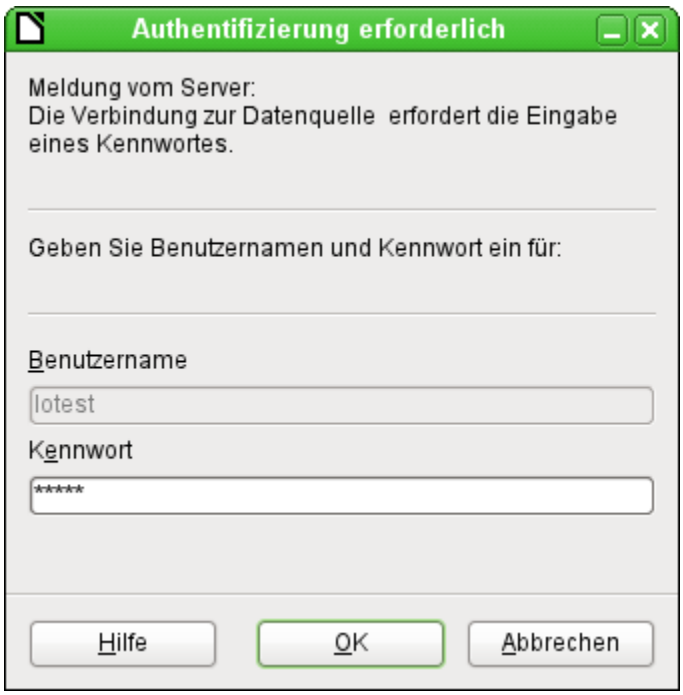



Der Datenbankname muss bekannt sein. Befindet sich der Server auf dem gleichen Rechner wie die Benutzeroberfläche, unter der die Datenbank erstellt wird, so kann als Server «localhost» gewählt werden. Ansonsten kann die IP-Adresse (z.B. 192.168.0.1) oder auch je nach Netzwerkstruktur der Rechnername oder gar eine Internetadresse angegeben werden. Es ist also ohne weiteres möglich, mit Base auf die Datenbank zuzugreifen, die vielleicht auf der eigenen Homepage bei irgendeinem Provider liegt.

Bei der Arbeit mit Base über das Internet sollte sich der Nutzer allerdings darüber bewusst sein, wie seine Verbindung zu der Datenbank gestaltet ist. Gibt es eine verschlüsselte Verbindung? Wie erfolgt die Passwortübertragung?

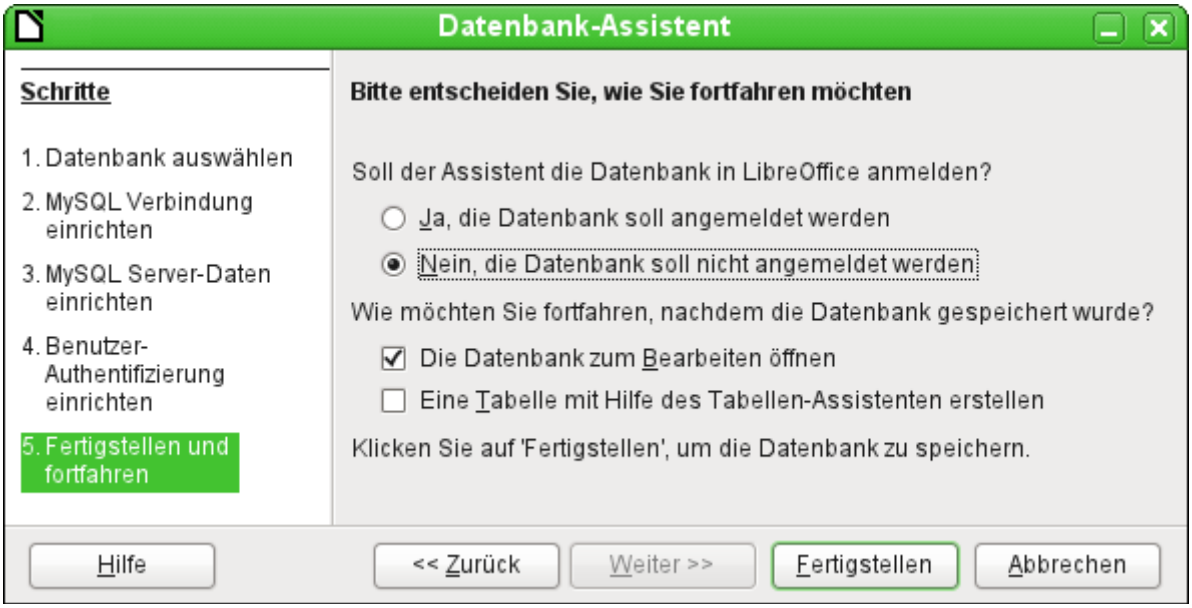



Jede über das Netz erreichbare Datenbank sollte mit einem Benutzernamen und einem Kennwort geschützt sein. Hier wird direkt getestet, ob die Verbindung klappt. Wichtig ist natürlich, dass der entsprechende Benutzer in MySQL bzw. der MariaDB entsprechend für den benannten Server eingerichtet wurde.



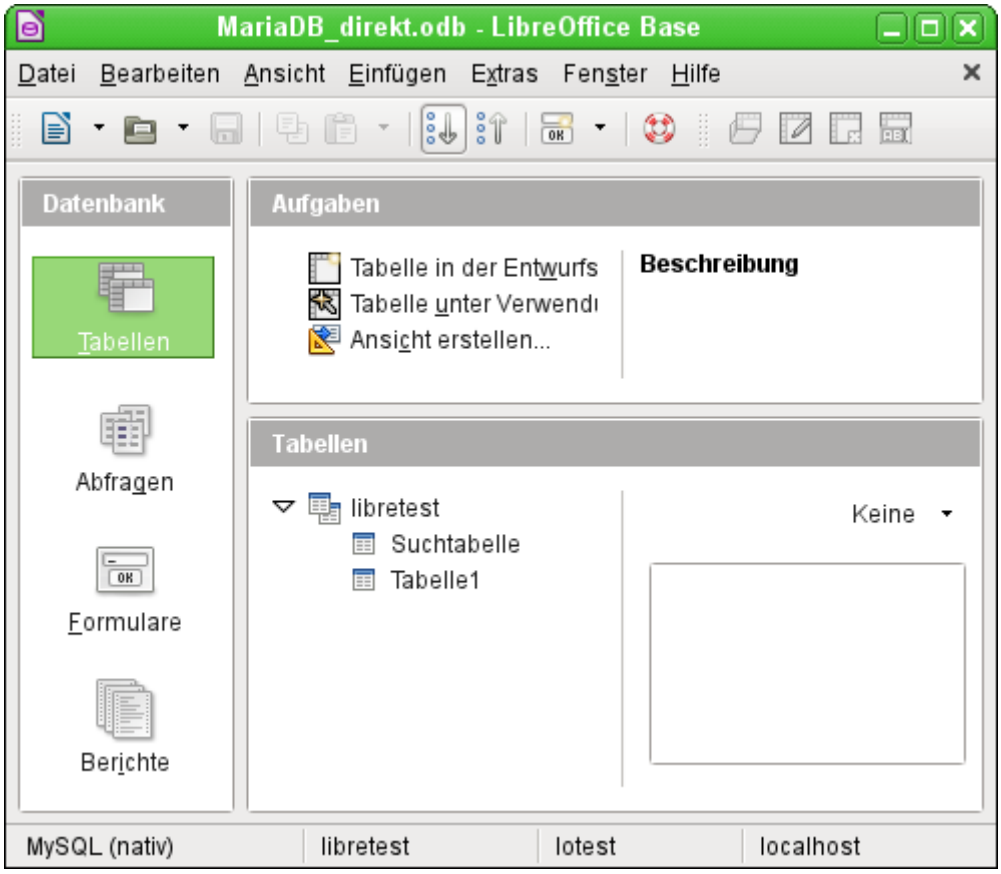

Dieses Fenster erscheint anschließend bei jedem Start der Datenbankdatei, wenn das erste Mal auf die MySQL-Datenbank zugegriffen wird.

Der Verbindungstest startet die Authentifizierung mit vorgegebenem Benutzernamen. Nach Passworteingabe erfolgt die Meldung, ob die Verbindung erfolgreich hergestellt werden kann. Läuft z.B. MySQL zur Zeit nicht, so kommt natürlich an dieser Stelle eine Fehlermeldung.



Auch hier wird die Datenbank nicht angemeldet, da sie nur zu ersten Tests aufgebaut wurde. Eine Anmeldung ist erst notwendig, wenn andere Programme wie z.B. der Writer für einen Serienbrief auf die Daten zugreifen sollen.

Der Assistent beendet die Verbindungserstellung mit dem Abspeichern der gewünschten Datenbankverbindung. In dieser *.odb-Datei liegen jetzt lediglich diese Verbindungsinformationen, die bei jedem Datenbankstart ausgelesen werden, so dass auf die Tabellen der MySQL-Datenbank zugegriffen werden kann.




*Ansicht der geöffneten Datenbankdatei mit Tabellenübersicht und in der Fußzeile der Benennung des verwendeten Treibers «MySQL (nativ)», des Datenbanknamens «libretest», des Nutzers der Datenbank «lotest» und des Servers, auf dem die Datenbank läuft, nämlich «localhost».*

Über «Fertigstellen» wird die Base-Datei erstellt und die Ansicht auf die Tabellen der MySQL-Datenbank geöffnet. Die Tabellen der Datenbank werden unter dem Namen der Datenbank selbst aufgeführt.

Beim aktuell lauffähigen Treiber von «aoo-my-sdbc» wird nur die Datenbank «test» angezeigt, für die auch die Verbindung bestimmt war. Vorherige Treiber von LO boten auch andere MySQL-Datenbanken auf dem gleichen Server zur Auswahl.

Auch mit dem Treiber «aoo-my-sdbc» ist aber ein Zugriff auf die anderen Tabellen z.B. für Abfragen möglich, sofern natürlich der angegebene Datenbanknutzer, im obigen Fall also «LibreOffice-Tester», mit seinem Passwort auf die Daten zugreifen kann. Im Gegensatz zu den bisherigen nativen LO-Treibern ist hier allerdings kein schreibender Zugriff auf andere Datenbanken des gleichen MySQL-Datenbankservers möglich.

Im Unterschied zu der internen Datenbank von Base taucht bei Abfragen entsprechend in MySQL für die Definition der Tabelle immer auch der Datenbankname auf, hier z.B.

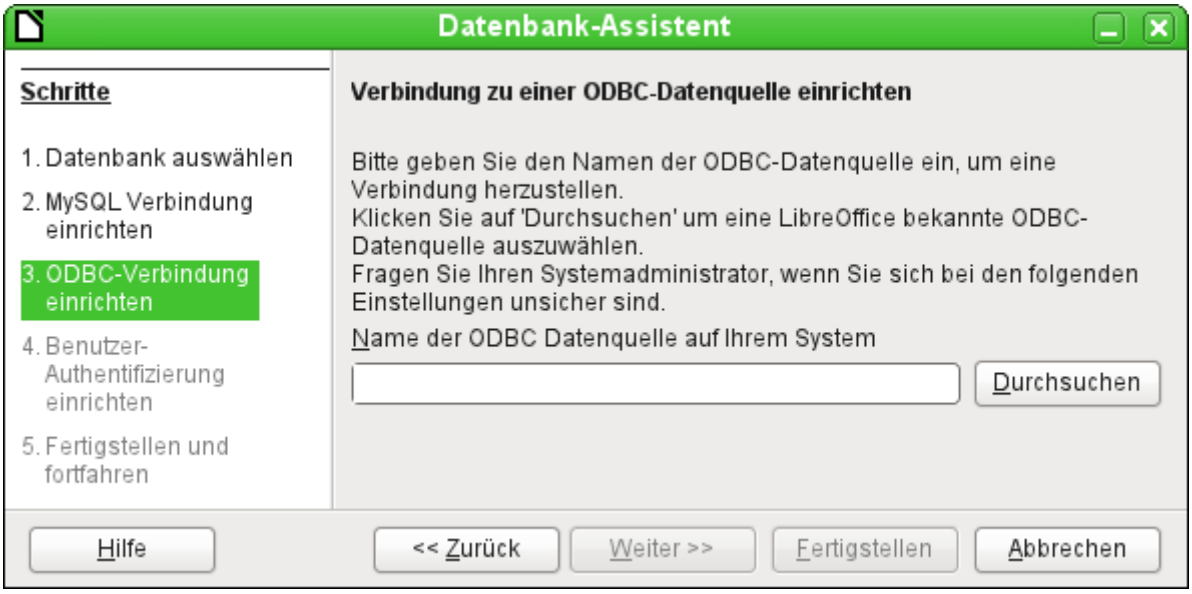
```
… FROM "test"."Klasse" AS "Klasse" ...
```

Hier ist es also auf jeden Fall notwendig, der Kombination aus Datenbankname und Tabellenname mit dem Zusatz «AS» einen alternativen Alias-Namen zuzuweisen. Genaueres siehe dazu in dem Kapitel «Verwendung eines Alias in Abfragen».

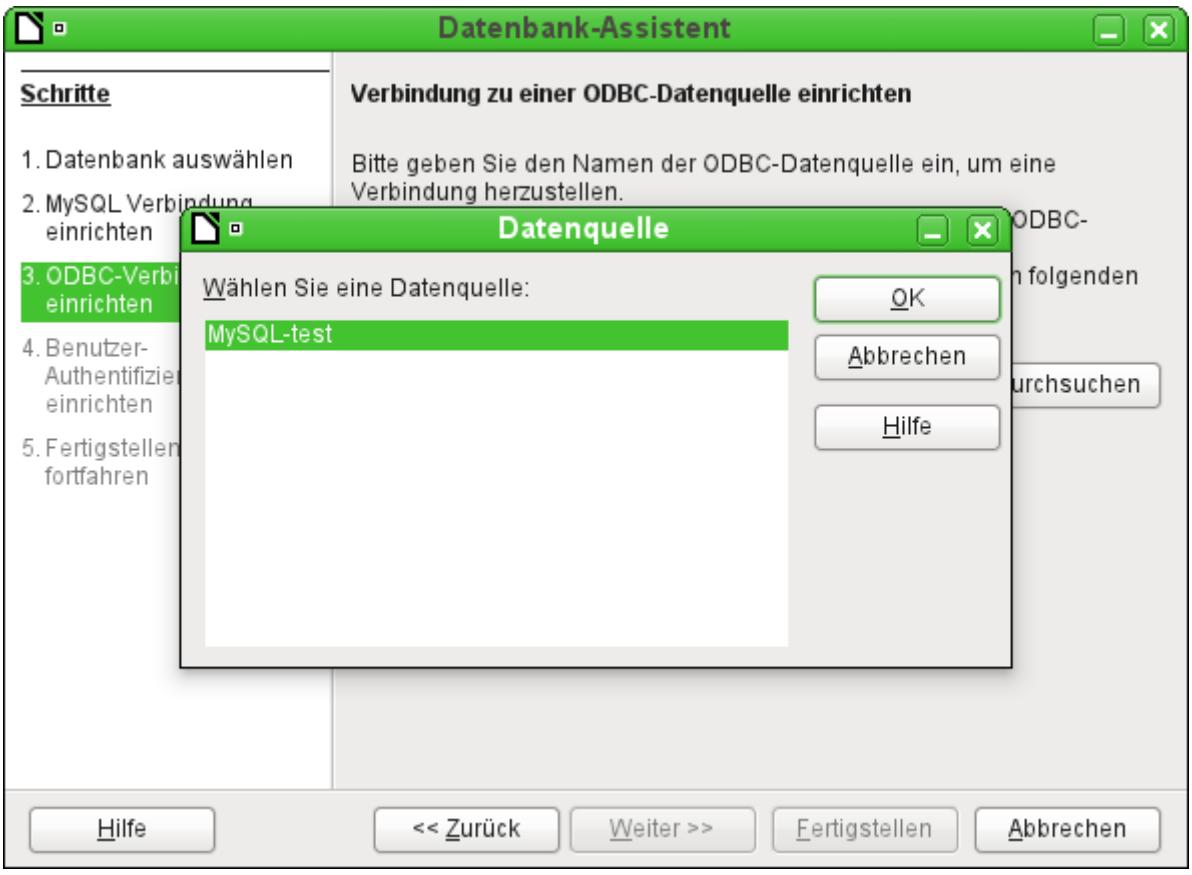
Tabellen können in der Datenbank erstellt und gelöscht werden. Automatisch hochzählende Zahlenfelder («Autowert») funktionieren und lassen sich auch bei der Tabellenerstellung auswählen. Sie starten bei MySQL mit dem Wert 1.

### Die ODBC-Verbindung

Die ersten Schritte zur ODBC-Verbindung sind gleich denen zur direkten Verbindung. Wird beim zweiten Schritt dann die ODBC-Verbindung für MySQL gewählt, dann erscheint das folgende Fenster des Datenbank-Assistenten:




Die ODBC Datenquelle hat nicht unbedingt den gleichen Namen wie die Datenbank in MySQL selbst. Hier muss der Name eingetragen werden, der auch in der Datei «odbc.ini» steht. Die einfachste Möglichkeit besteht hier darin, über den Button Durchsuchen den Namen aus der «odbc.ini» direkt auszulesen.

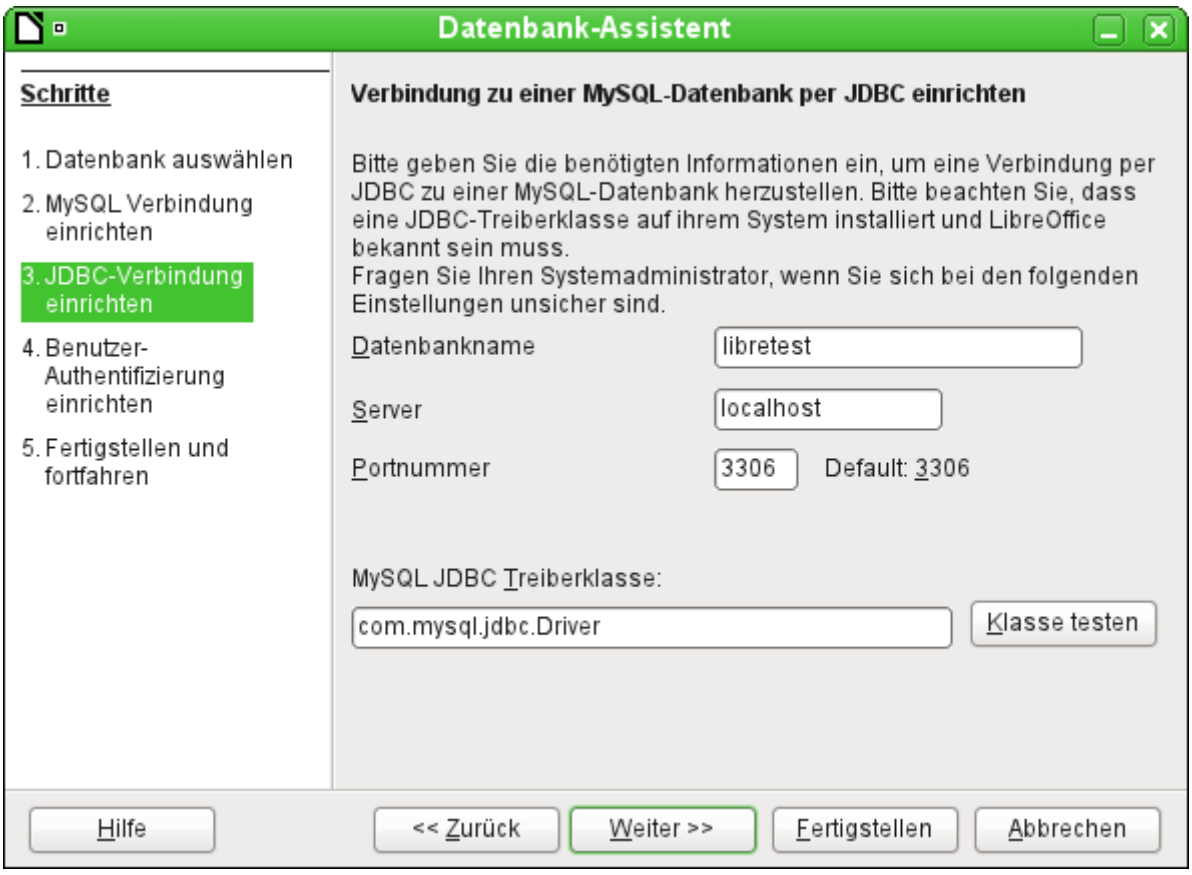



Beim Test erscheint der Name aus der «odbc.ini». Auch wenn hier wieder nur mit einer Datenbank verknüpft wird, können andere Tabellen des MySQL-Servers sehr wohl gelesen werden.

Schritt 4 und 5 laufen wieder identisch zur Direktverbindung ab.

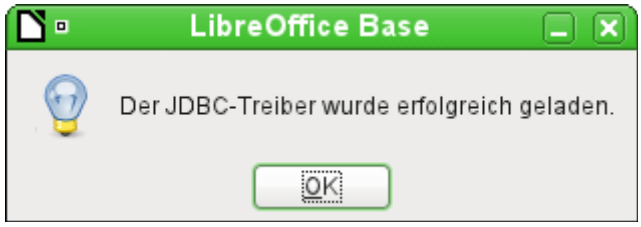
### Die JDBC-Verbindung

Auch bei der JDBC-Verbindung sind die ersten Schritte gleich hier unterscheidet sich wieder nur Schritt 3 von den anderen Schritten des Assistenten:




Der Assistent fragt hier die gleichen Informationen ab wie bei der direkten Verbindung. Der Datenbankname ist der, der auch in MySQL selbst verwandt wird.

Über «Klasse testen» wird überprüft, ob das Archiv mysql-connector-java.jar über Java erreichbar ist. Entweder muss dieses Archiv im Pfad der ausgewählten Java-Version liegen oder direkt in LibreOffice eingebunden werden.




Alle weiteren Schritte sind wieder identisch zu denen der vorherigen Verbindungen. Die Verbindungen zu anderen Datenbanken des gleichen MySQL-Datenbankservers funktionieren ebenfalls nur lesend.

## PostgreSQL

Für die PostgreSQL-Datenbank gibt es einen direkten Treiber von LO, der von vornherein mit installiert wird. Damit der Kontakt auch sicher passt zuerst aber eine kurze Anleitung zu den ersten Schritten nach der Installation von PostgreSQL.

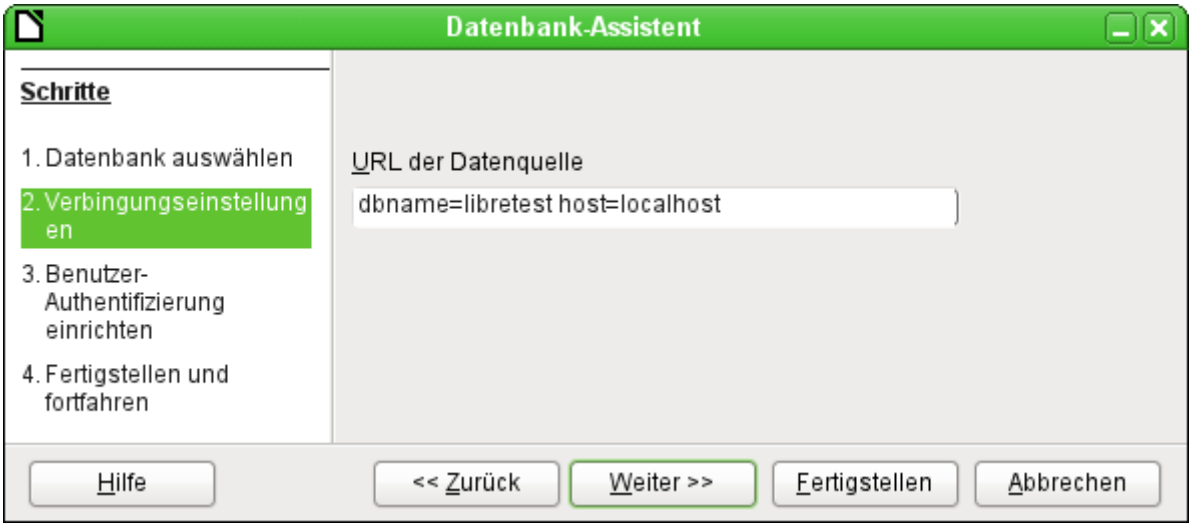
### *Erstellen eines Nutzers und einer Datenbank*

Die folgenden Schritte sind nach einer Installation über den Paketmanager in OpenSUSE erforderlich. Sie ähneln vermutlich denen unter anderen Betriebssystemen.

- Dem Nutzer «postgres» muss zuerst ein Passwort zugewiesen werden. Das kann mit dem Administartionstool des Betriebssystemns erfolgen.
- Der Postgre-Server muss gestartet werden.
- Der Nutzer «postgres» muss sich mit
  **`su postgres`**
  auf der Konsole anmelden.
- Ein einfacher Datenbanknutzer, hier «lotest» sollte erstellt werden:
  **`createuser -E -P lotest`**
- Damit der Datenbanknutzer anschließend auch auf die zu gründende Datenbank zugreifen kann, muss in der Datei /var/lib/pqsql/data/pg_hba.conf ein Eintrag geändert werden. In dieser Datei werden u.a. die Methoden zur Identifizierung der Nutzer auf verschiedenen Ebenen festgelegt. Die Methode, mit der LO-Base kommunizieren kann, ist die Methode «password», nicht, wie voreingestellt, die Methode «ident».
- Die Datenbank «libretest» für den Nutzer «lotest» wird erstellt:
  **`createdb -h localhost -U lotest -W -e libretest`**

### *Direkte Verbindung zu Base*

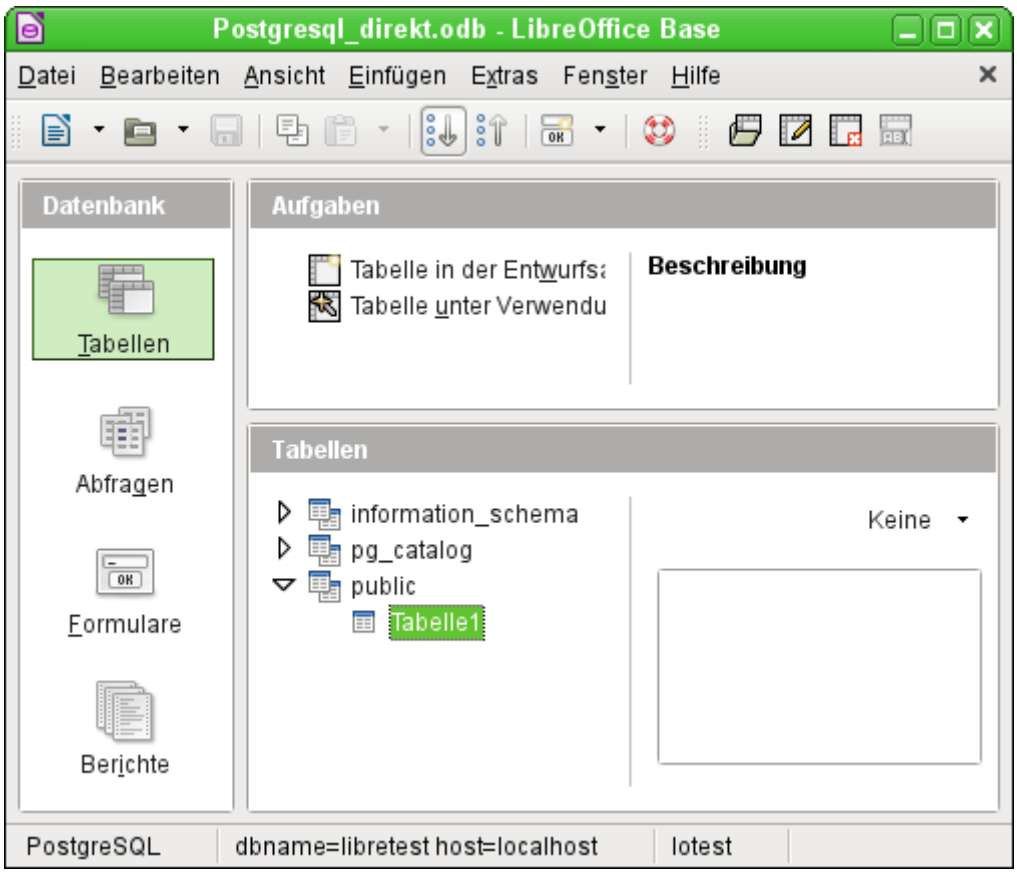
Im Assistenten wird der Eintrag «postgres» ausgewählt. Die Verbindungseinstellung geschieht über die Angabe des Datenbanknamens («dbname») und des Hostes:




Die Benutzer-Authentifizierung sieht genauso aus wie die unter MySQL.

Wird die Datenbank das erste Mal geöffnet, so erscheinen für den User erst einmal eine Unmenge an Tabellen aus dem Bereich «information_schema». Diese Tabellen sind ebenso wie die im Bereich «pg_catalog» für den Nutzer nur lesbar, nicht schreibbar. Diese verschiedenen Bereiche bezeichnet PostgreSQL als «Schema». Der Nutzer legt neue Tabellen im Schema «public» an.

| **Hinweis** | Werden Tabellen z.B. aus der internen HSQLDB nach PostgreSQL kopiert, so schlägt der Importassistent den einfachen Tabellennamen aus der HSQLDB, z.B. «Tabelle1», vor. Der Import mit diesem Namen schlägt allerdings fehl. Stattdessen muss vor den Namen die Schemabezeichnung ergänzt werden: aus «Tabelle1» wird «public.Tabelle1». |
|---|---|

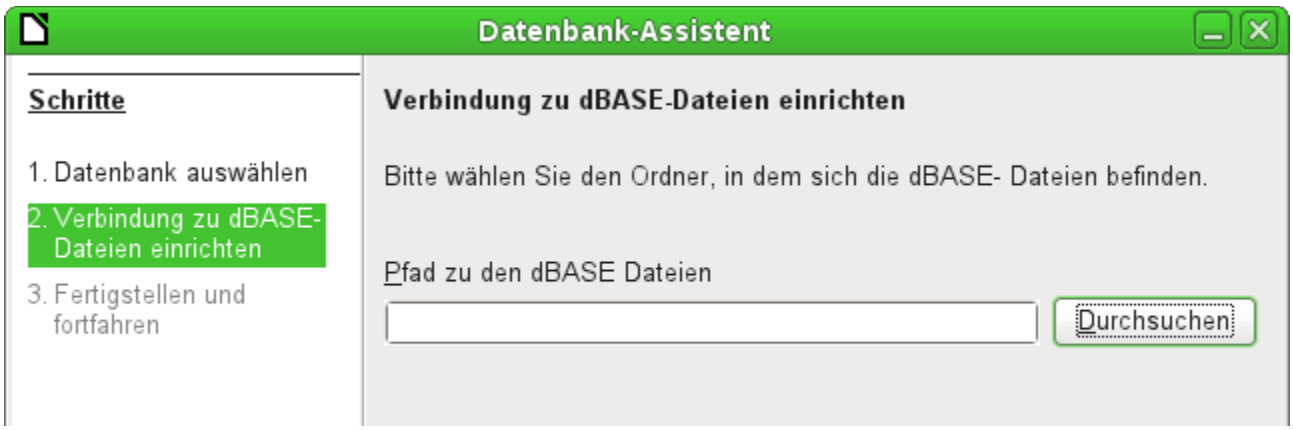



Bei der Erstellung von Tabellen kann es vorkommen, dass Base Datentypen vorschlägt, die die aktuelle PostgreSQL-Installation nicht verarbeiten kann. So werden standardmäßig die Felder mit dem Feldtyp «Text [character_data]» vorgeschlagen. Den kann PostgreSQL aber nicht verarbeiten. Eine Änderung auf den Typ «Text [varchar]» schafft hier Abhilfe.

## dBase-Datenbanken

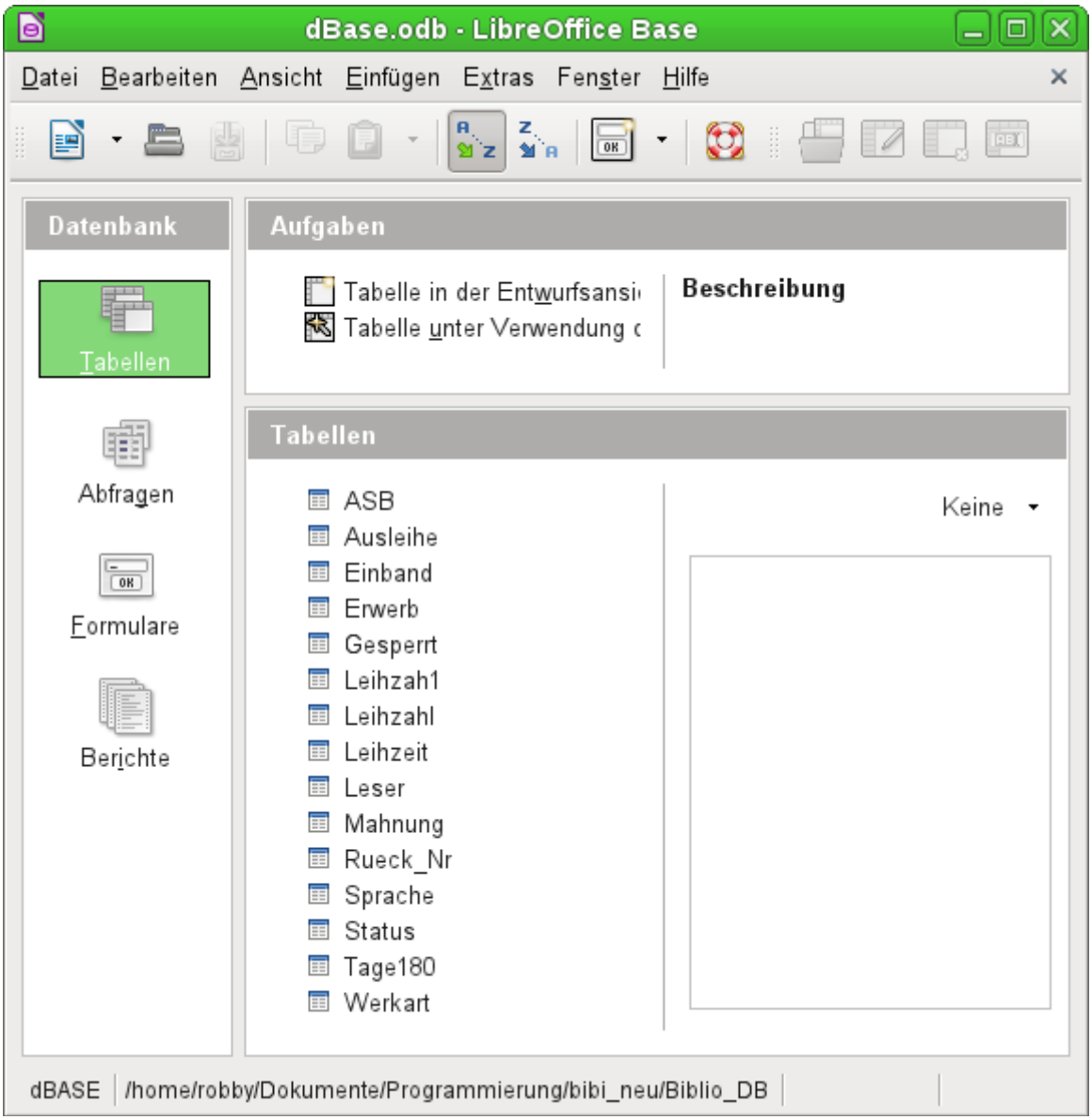
Bei dBase handelt es sich um ein Datenbankformat, dass alle Daten in einem beliebigen Verzeichnis in separaten Tabellen bereitstellt. Verknüpfungen zwischen den Tabellen müssen über die Programmstruktur erledigt werden. Dbase hat keine Möglichkeit, intern zu verhindern, dass z.B. in der Medien-Datenbank ein Medium gelöscht wird, der Verweis darauf aber weiterhin in der Tabelle zum Medienentleih erhalten bleibt.

| **Hinweis** | Zur Zeit werden nur dBase-Dateien berücksichtigt, die mit der in Kleinbuchstaben geschriebene Dateiendung *.dbf versehen sind. Andere Endungen wie z.B. *.DBF werden nicht ausgelesen. (*Bug 46180*) |
|---|---|



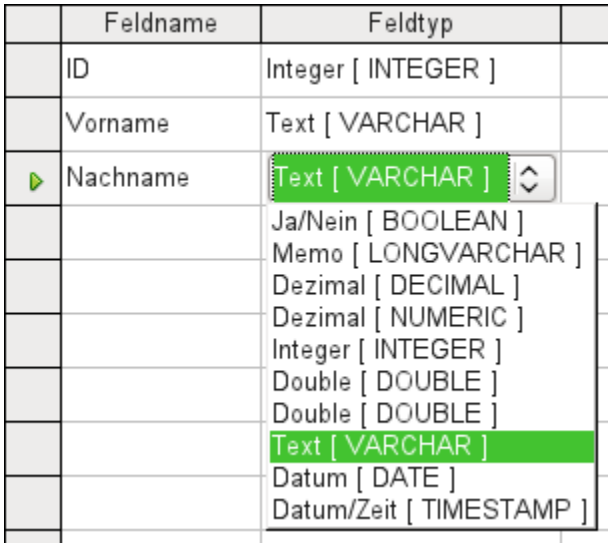

Die Verbindung wird zu einem Verzeichnis hergestellt. Alle *.dbf-Dateien, die sich in diesem Verzeichnis befinden, werden anschließend angezeigt und können über Abfragen oder Formulare miteinander verbunden werden.

Tabellen in dBase haben kein unverwechselbares Feld für jeden Datensatz («Primärschlüssel»). Sie können also vom Prinzip her so beschrieben werden wie Tabellenblätter in Calc.



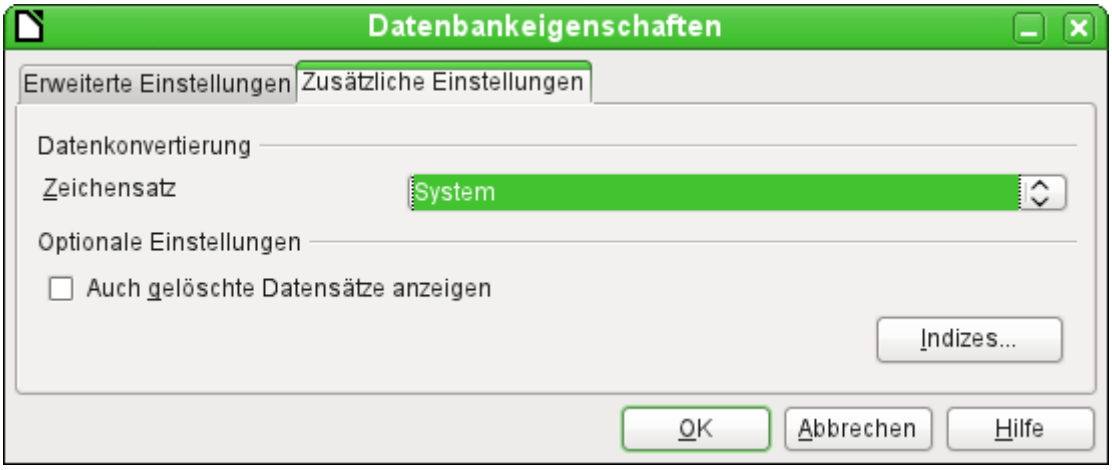
Tabellen können in Base neu erstellt werden. Sie werden dann einfach als neue Dateien in das angegebene Verzeichnis geschrieben.

Für neue Tabellen stehen deutlich weniger Feldtypen zur Verfügung als für die interne HSQLDB. Dabei sind noch einige Bezeichnungen der folgenden Abbildung als Verdoppelungen anzusehen.




Dbase bietet sich besonders an, wenn es um Weitergabe und vielfältige Verarbeitungsmöglichkeit von Daten geht. Schließlich können auch Tabellenkalkulationen dBase-Tabellen direkt einlesen.

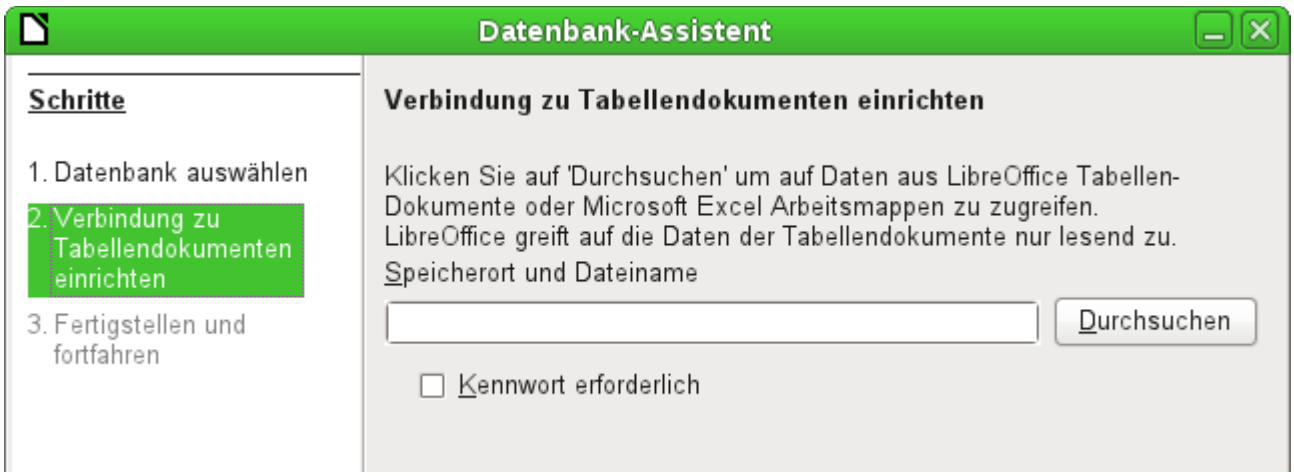
Base übernimmt einfach die Codierung, die dem Betriebssystem entspricht. Alte dBase-Dateien weisen dadurch leicht Fehler beim Import von Sonderzeichen auf. Der Zeichensatz kann anschließend über **Bearbeiten** → **Datenbank** → **Eigenschaften** → **Zusätzliche Einstellungen** entsprechend korrigiert werden:




| **Hinweis** | Der Importassistent für dBase hat Probleme, numerische Feldtypen und Ja/Nein-Felder automatisch richtig zu erkennen. (*Bug 53027*) Hier muss entsprechend nachgebessert werden. |
|---|---|

## Tabellendokumente

Tabellenquellen für Datenbanken können auch Tabellendokumente sein. Wird aber eine Calc-Tabelle als Grundlage genommen, so ist jedes Editieren der Tabelle ausgeschlossen. Selbst wenn das Calc-Dokument auch noch geöffnet wird, wird dort ein Schreibschutz gesetzt.

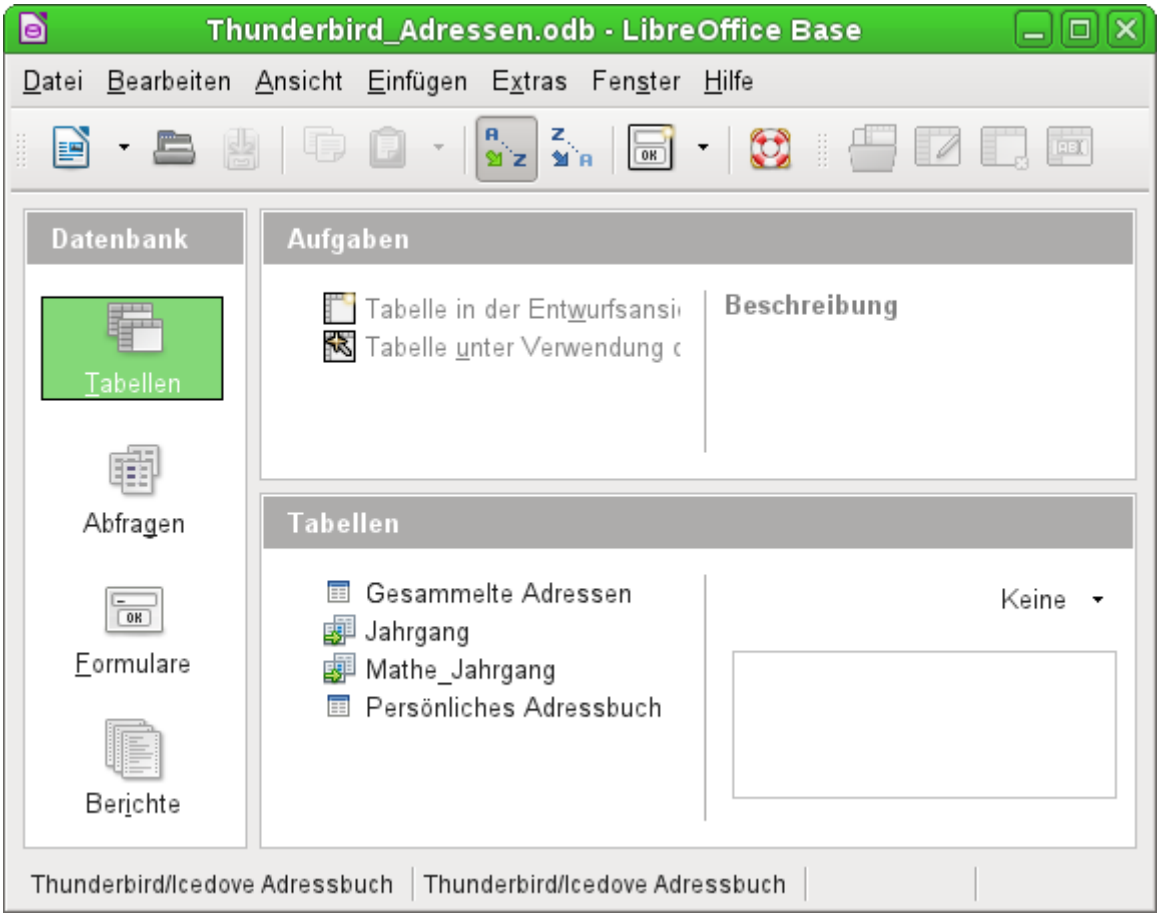



Die einzige Abfrage ist letztlich, an welcher Position das Tabellendokument liegt und ob die Tabelle passwortgeschützt ist. Base öffnet dann das Tabellendokument und übernimmt alle Tabellenblätter mit ihrem Inhalt. Aus der ersten Zeile bildet Base die Feldbezeichnungen, aus den Tabellenblattbezeichnungen die Tabellenbezeichnungen.

Beziehungen zwischen den Tabellen lassen sich in Base nicht erstellen, da Calc nicht Grundlage für eine relationale Datenbank ist. Auch lassen sich Abfragen nicht über eine Tabelle hinaus erstellen.

## Thunderbird Adressbuch

Die Verbindung zu einem Adressbuch wie z.B. dem Thunderbird-Adressbuch sucht der Assistent automatisch. Er fragt lediglich nach einem Speicherort für die *.odb-Datei, die er erstellen soll.



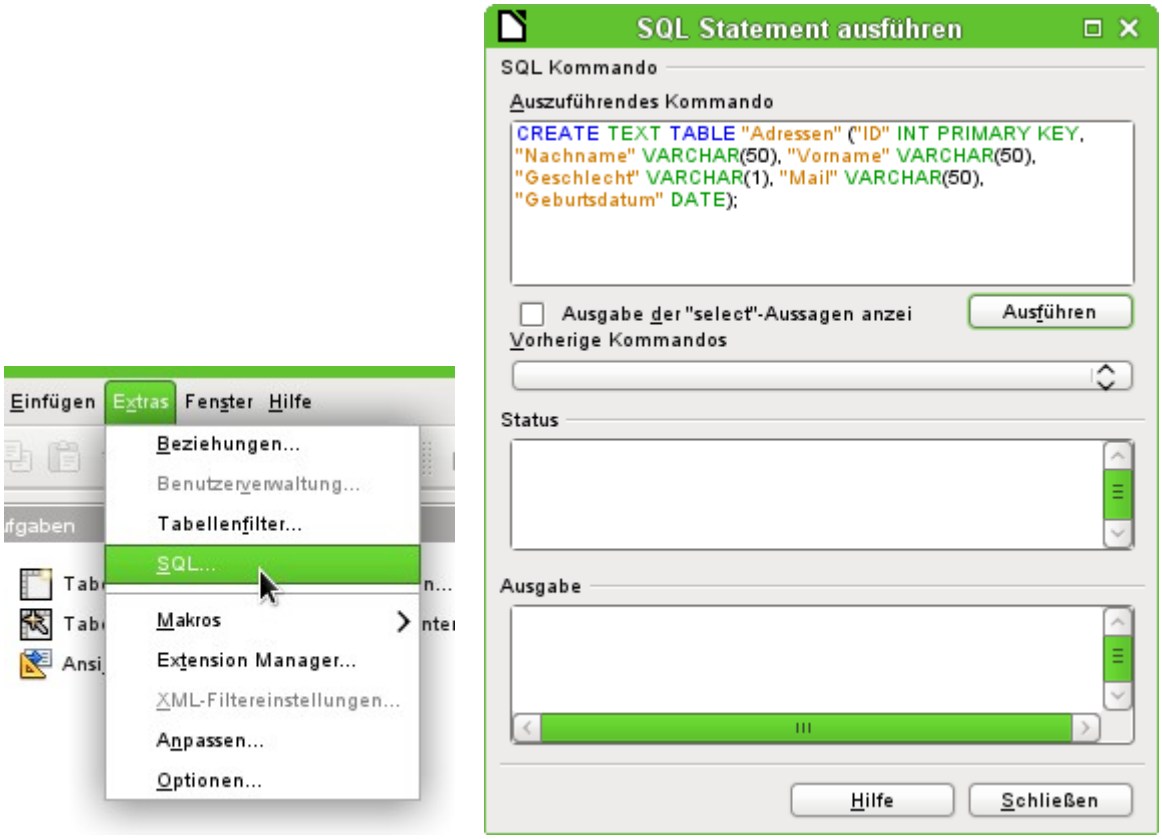
Alle Tabellen werden dargestellt. Wie in Calc sind die Tabellen aber nicht editierbar. Base macht die Adressen damit nur für Abfragen und z. B. für die Verwendung in Serienbriefen zugänglich.

| Hinweis | Seit der Version LO 4.* wird unter Linux und MacOSX nur das Adressbuch «Persönliches Adressbuch» als «AddressBook» ausgelesen. Dadurch werden die Daten, die standardmäßig im Adressbuch «Gesammelte Adressen» liegen, nicht zugänglich. (*Bug 64741*)<br>In der Version LO 4.2 ist dieser Bug behoben. Allerdings sind zusammengefasste Gruppen zur Zeit nur als Teil des «persönlichen Adressbuches» sichtbar. Gruppen aus «gesammelte Adressen» werden nicht dargestellt. |
|---|---|

## Texttabellen

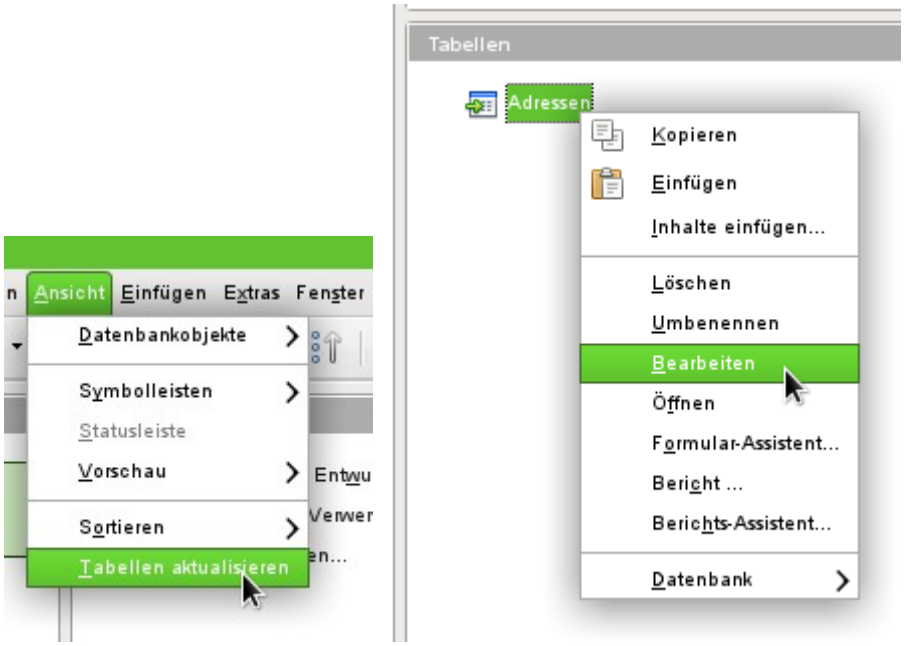
Ein gängiges Austauschformat zwischen Datenbanken ist das *.csv-Format. Daten werden hier in einer durch einfache Textbearbeitungsprogramme les- und schreibbaren Form gespeichert. Einzelne Felder werden durch Kommata getrennt. Enthält ein Feld Text mit einem Komma, so werden die Textfelder mit doppelten Anführungszeichen versehen. Ein neuer Datensatz beginnt nach einem Zeilenumbruch.

Soll zum Beispiel der Inhalt eines Adressbuches, das nicht direkt von den Treibern von Base unterstützt wird, eingelesen werden, so kann dies entweder über einen Import einer solchen *.csv-Datei erfolgen (hierzu ist gegebenenfalls ein Zwischenschritt über Calc notwendig) oder die Datei wird direkt als Texttabelle in die Datenbank eingefügt. Um dort bearbeitet werden zu können benötigt die *.csv-Datei allerdings ein Feld, das als eindeutiges Feld («Primärschlüssel») erkennbar ist.

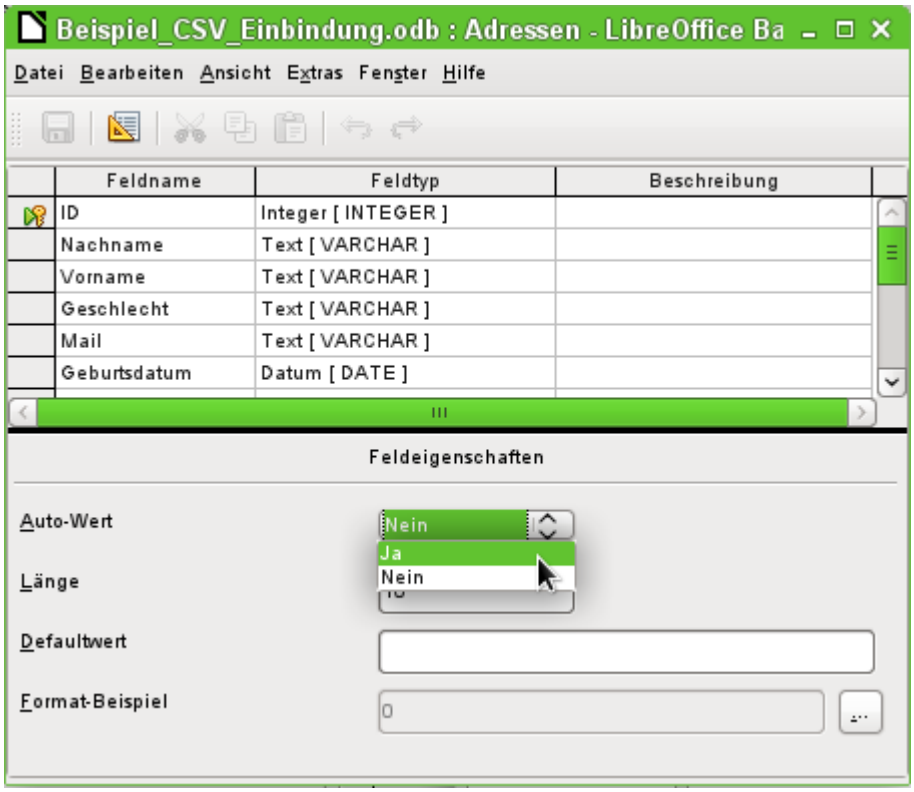



Der Zugriff zu einer Texttabelle kann nicht über die grafische Benutzeroberfläche erstellt werden. Stattdessen wird über Extras → SQL eine Texttabelle erstellt. Die Felder in der Texttabelle müssen dem entsprechen, was die Texttabelle in der entsprechenden Reihenfolge liefert. Das Feld "ID"

muss z. B. positive Ganzzahlen enthalten, das Feld "Geburtsdatum" z. B. ein Datum in der Schreibweise Jahr – Monat – Tag.




Die Tabelle ist nicht direkt auf der Benutzeroberfläche sichtbar. Falls jetzt noch eine Ergänzung erfolgen soll muss über Ansicht → Tabellen aktualisieren die Tabelle verfügbar gemacht werden. Das Symbol der Tabelle macht deutlich, dass es sich nicht um eine «normale» Tabelle der Datenbank handelt.




Die Tabelle wurde zum Bearbeiten geöffnet und das Primärschlüsselfeld auf einen automatisch hochzählendes Feld umgestellt.

Über Extras → SQL muss jetzt noch die Verbindung zur externen Texttabelle erstellt werden. Diese Texttabelle liegt im gleichen Verzeichnis wie die Datenbankdatei selbst.

```
SET TABLE "Adressen" SOURCE "Adressen.csv;encoding=UTF-8";
```

Anschließend steht die Texttabelle ganz normal zur Eingabe zur Verfügung. Allerdings ist hierbei Vorsicht geboten:

- Die Texttabelle kann gleichzeitig auch von externen Textprogrammen oder Tabellenkalkulationen geöffnet und bearbeitet werden. Datenverlust ist dann nicht auszuschließen.
- Änderungen an bereits geschriebenen Datensätzen führen dazu, dass die entsprechende Zeile in der Originaldatei geleert wird und in der neuen Fassung an den Schluss der Tabelle angefügt wird. Die obige Ansicht präsentiert z.B. sauber 4 beschriftete Zeilen mit richtig sortierter ID. Die Originaldatei zeigt hingegen nach einer Änderung im 2. Datensatz die folgende Reihenfolge, geordnet nach der ID: 1, Leerzeile, 3, 4, 2

Für die Verbindung zur Textdatei stehen verschiedene Parameter zur Verfügung.

```
SET TABLE "Adressen" SOURCE
"Adressen.csv;ignore_first=false;all_quoted=true;encoding=UTF-8";
```

Mit «ignore_first=true» wird die erste Zeile der Datei nicht eingelesen. Dies ist sinnvoll, wenn es sich dabei um die Feldbezeichnungen handelt. Der interne Standardwert der HSQLDB steht hier auf «false».

Standardmäßig werden von der HSQLDB Texte nur in doppelte Anführungszeichen gesetzt, wenn sie ein Komma enthalten, da das Komma die Standardtrennung der Felder ist. Soll jeder Text in doppelte Anführungzeichen gefasst werden, so ist «all_quoted=true» zu setzen.

Zu weiteren Parametern siehe: *http://hsqldb.org/doc/guide/ch09.html#set_table_source-section* .

Mit

```
SET TABLE "Adressen" READONLY TRUE;
```

wird schließlich davor geschützt, dass in die Tabelle überhaupt etwas geschrieben wird. So steht dann die Tabelle als normale Adresstabelle wie eine Tabelle aus Adressbüchern z.B. von Mailprogrammen schreibgeschützt zur Verfügung. Die gesonderte Erstellung eines Schreibschutzes ist allerdings nur notwendig, wenn für die Tabelle erst einmal ein Primärschlüssel erstellt wurde.

## Firebird

Die etwas in die Jahre gekommene alte HSQLDB-Version der internen Datenbank soll durch eine interne Firebird-Datenbank ersetzen werden. Zur Version LO 4.2.* ist die interne Firebird-Datenbank erst einmal nur als «experimentelle Funktion» zusätzlich wählbar. Das liegt einfach daran, dass längst noch nicht alle Funktionen richtig integriert wurden. Um aber zu sehen, was Firebird leiten kann, hier jetzt also die Verbindung zu einer externen Firebird-Datenbank zu testen.

Da die Dokumentation dazu nicht ganz so umfangreich ist wie z.B. zu MySQL oder PostgreSQL hier erst einmal die wichtigsten Schritte bei der Installation.

### *Erstellen eines Nutzers und einer Datenbank*

Unter Linux stehen für Firebird entsprechende Pakete in der Paketauswahl bereit. Hier muss nach der Installation eigentlich nur noch der Server eingeschaltet werden. Die folgenden Schritte haben unter OpenSUSE 12.3 zu einer funktionierenden Datenbank mit Firebird geführt:

- **`sysdba`** ist der Nutzername für den Admin-Account. Das Standardpasswort ist **`masterkey`**. Dies sollte in Produktivumgebungen geändert werden.
- Notwendig für die Installationsschritte bis zur funktionierenden Datenbank sind Administratorrechte auf dem Rechner. Unter Linux reicht der SuperUser **`su.`**
- Eine Passwortänderung geht im Terminal über
  **`gsec -user sysdba -pass masterkey -mo sysdba -pw neuespasswort`**
- Ein neuer Nutzer sollte gegründet werden - hier gezeigt mit altem Standardpasswort des sysdba:
  **`gsec -user sysdba -pass masterkey -add lotest -pw libre`**
  Damit ist ein neuer Nutzer mit dem Namen **`lotest`** und dem Passwort **`libre`** erstellt.
- Danach wird, immer noch als SuperUser, eine Datenbank für den neuen Nutzer erstellt. Hierzu wird das Hilfsprogramm **`isql-fb`** aufgerufen.
  **`isql-fb`**
  Es erscheint die folgende Meldung:
  **`Use CONNECT or CREATE DATABASE to specify a database`**
  direkt gefolgt von der Eingabeaufforderung **`SQL>.`** Hier entsprechend passende Einträge:
  **`SQL>CREATE DATABASE 'libretest.fdb'`**
  **`CON>user 'lotest' password 'libre';`**

Alle Eingaben wurden in der Funktion als Systemadministartor Root getätigt. Damit ist die Datenbank nicht dem richtigen Benutzer für den Netzwerkbetrieb zugeordnet.

Die Datenbank muss als Nutzer und als Gruppe «firebird» eingetragen bekommen. Sonst klappt der Kontakt später nicht.

### *Firebird-Verbindung über JDBC*

Zuerst muss das Jar-Archiv in LO eingebunden werden. Irritierend ist, dass es kein Archiv mit einer Bezeichnung firebird-*.jar gibt. Der aktuelle JDBC-Treiber ist auf der Seite *http://www.firebirds-ql.org/en/jdbc-driver/* zu finden. Der Treiber beginnt mit der Bezeichnung Jaybird...

Aus der *.zip-Datei muss nur das Archiv jaybird-full-2.2.3.jar (oder aktuellere Version) kopiert und entweder in den Java-Pfad der Installation gelegt oder als Archiv direkt in LO eingebunden werden. Siehe dazu auch die entsprechende Einbindung bei MySQL.

Für die Angaben bei der JDBC-Installation sind folgende Parameter wichtig:

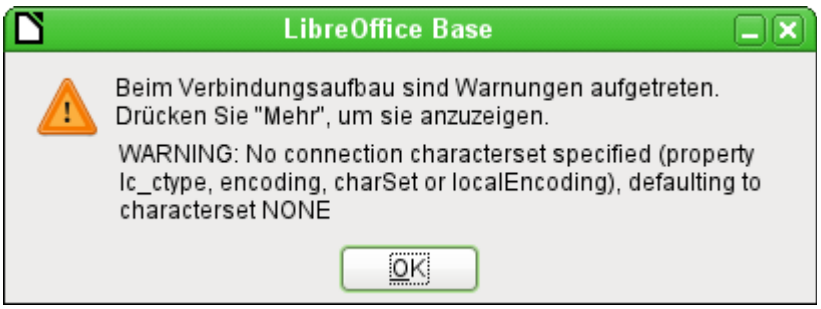
Treiberbezeichnung: **`org.firebirdsql.jdbc.FBDriver`**

JDBC URL **`jdbc:firebirdsql://host[:port]/<path_or_alias>`**

In obigen Beispiel wurde dies zu der URL

jdbc:firebirdsql://localhost/libretest.fdb?charSet=UTF-8

Ohne Zeichensatzangabe erscheint die folgende Fehlermeldung:




Bei der Erstellung von Tabellen muss darauf geachtet werden, dass die Formatierung der entsprechenden Felder («Feldeigenschaften») von Beginn an stimmt. Bei allen Zahlenwerten setzt sonst LO als Default-Wert merkwürdigerweise ein Währungsfeld ein.

Nachträgliche Änderungen z.B. der Feldeigenschaften von Feldern einer Tabelle sind nicht möglich, wohl aber die Erweiterung der Tabelle oder das Löschen der Felder.

### *Firebird-Verbindung über ODBC*

Aus dem Netz muss zuerst der passende ODBC-Treiber heruntergeladen werden: *http://www.firebirdsql.org/en/odbc-driver/*. Dieser Treiber besteht meist aus einer einfachen Datei, der libOdbcFb.so. Diese Datei wird meist mit Administratorrechten in einen vom System allgemein zugänglichen Pfad gelegt. Sie muss ausführbar sein.

In den für das System notwendigen Dateien odbcinst.ini und odbc.ini müssen Einträge ähnlich den folgenden existieren:

**odbinst.ini**

```
[Firebird]
Description = Firebird ODBC driver
Driver64 = /usr/lib64/libOdbcFb.so
```

**odbc.ini**

```
[Firebird-libretest]
Description     = Firebird database libreoffice test
Driver = Firebird
Dbname = localhost:/srv/firebird/libretest.fdb
SensitiveIdentifier = Yes
```

Diese beiden Dateien liegen bei einem Linux-System im Verzeichnis /etc/unixODBC. Die Variable SensitiveIdentifier muss auf jeden Fall auf «Yes» stehen, damit die Eingabe in Tabellen funktioniert, deren Namen und Feldbezeichnungen nicht in Großbuchstaben angegeben sind.

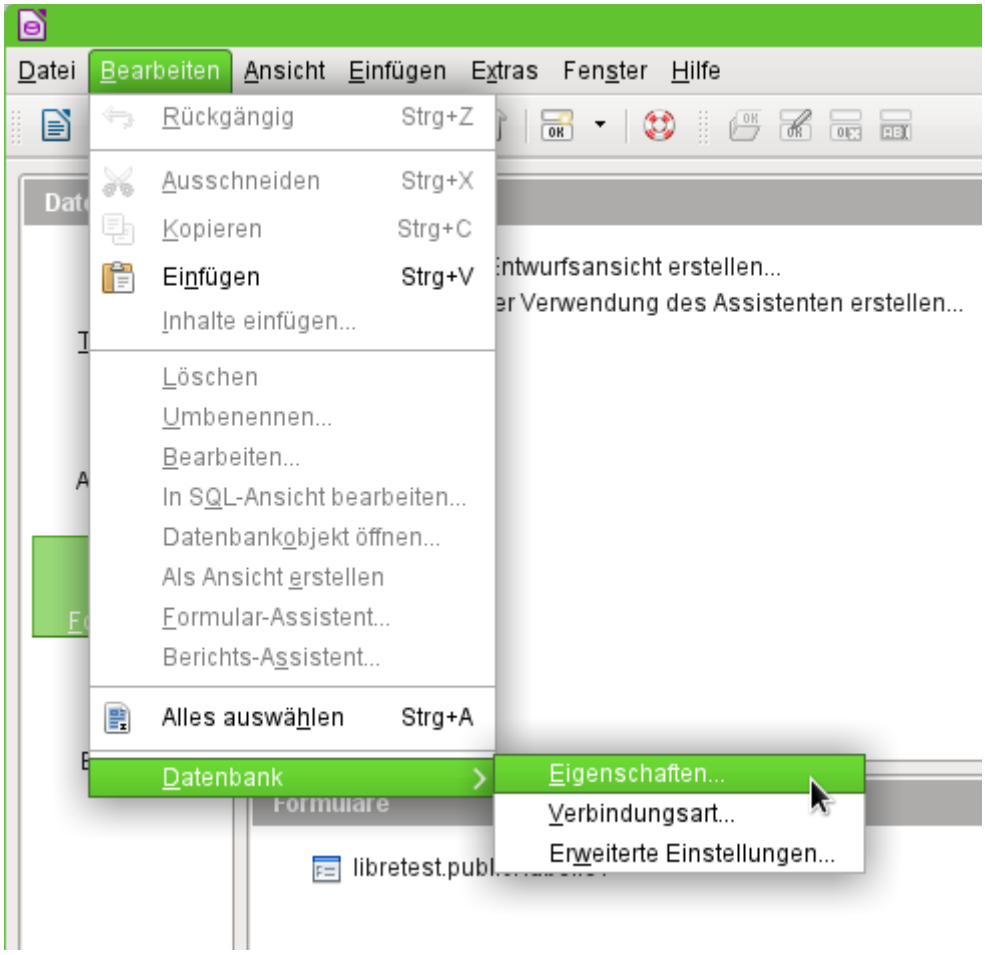
Zur Erstellung der Verbindung von LO mit der entsprechenden Datenquelle siehe *Die ODBC-Verbindung.*

Die ODBC-Verbindung ist leider zur Zeit nur sehr eingeschränkt funktionstüchtig. Beim Erstellen von Tabellen stürzt LO ab. Beim Eingeben von Werten in ein Integer-Feld wird bei Linux-64-bit-Versionen an die Datenbank ein leeres Feld weitergegeben ( *http://tracker.firebirdsql.org/browse/ODBC-141* ). Hier klappt die Zusammenarbeit mit dem ODBC-Treiber nicht. Da funktioniert der JDBC-Treiber eindeutig besser.

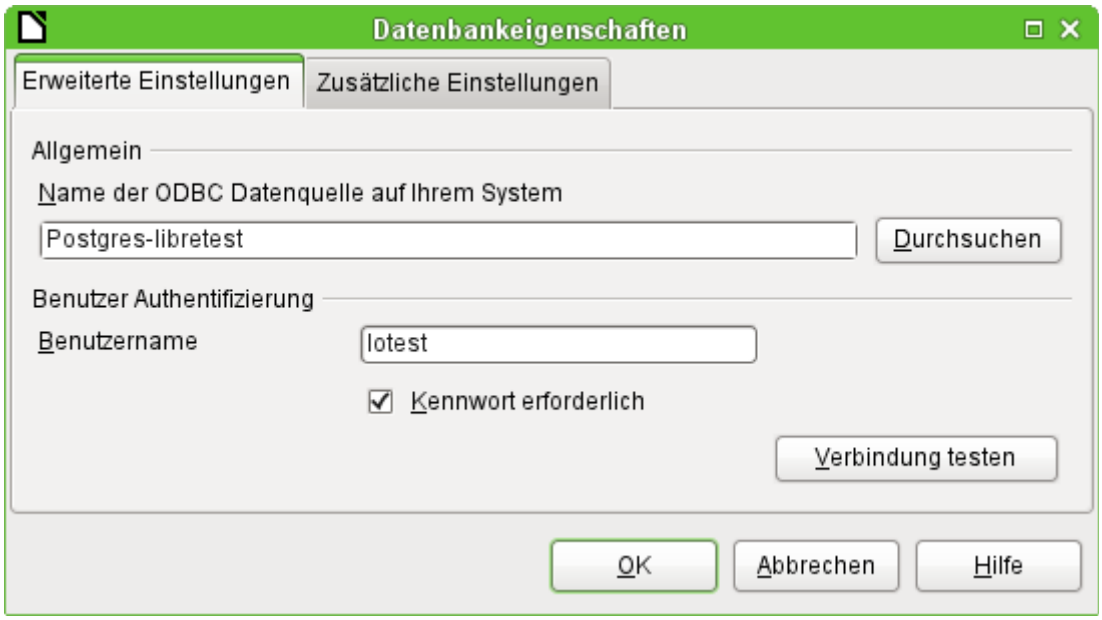
## Nachträgliche Bearbeitung der Verbindungseigenschaften

Vor allem bei Verbindungen zu externen Datenbanken kann es hin und wieder vorkommen, dass eine Verbindung nicht wie gewünscht einwandfrei funktioniert. Mal ist der Zeichensatz nicht korrekt, ein anderes Mal funktionieren Unterformulare nicht einwandfrei oder es ist einfach etwas an den grundsätzlichen Verbindungsparametern zur Datenbank geändert worden.

Die folgenden Screenshots zeigen die Möglichkeit auf, wie z.B. die Verbindungsparameter zu einer externen PostgreSQL-Datenbank im Nachhinein geändert werden können.

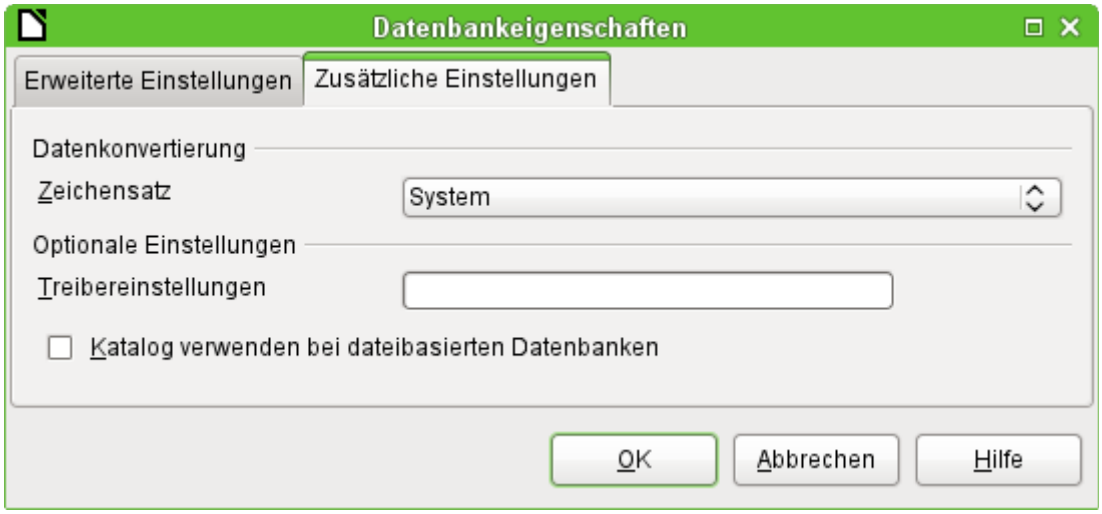



Über **Bearbeiten → Datenbank** werden die Einstellungsmöglichkeiten «Eigenschaften», «Verbindungsart» und «Erweiterte Einstellung» aufgerufen.



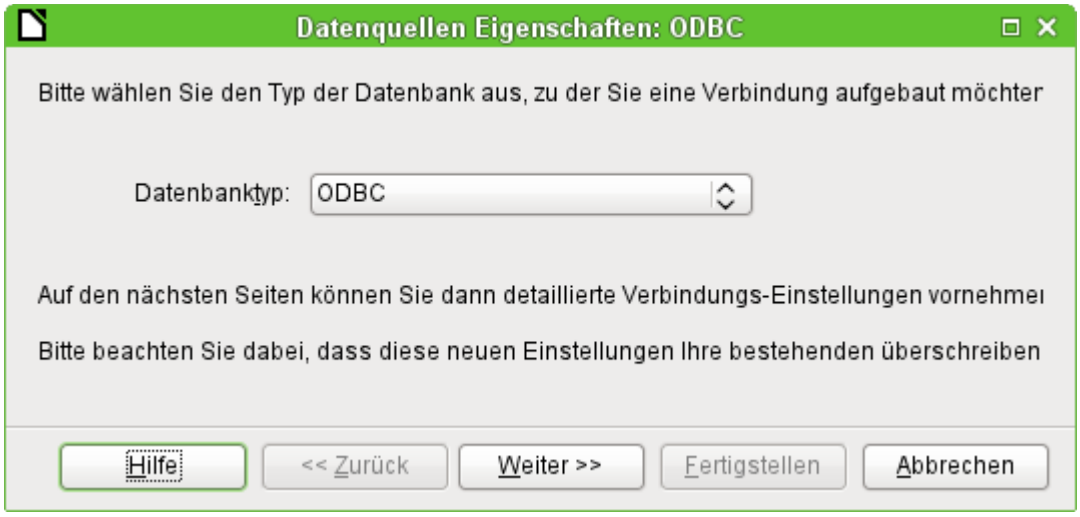

Falls sich der Name der Datenquelle geändert hat, kann dies hier geändert werde. Bei einer Verbindung über ODBC wird ja der Name, mit dem die Datenbank aufgerufen werden kann, in der

odbc.ini festgelegt. Der Name ist in der Regel nicht gleich dem eigentlichen Datenbanknamen in PostgreSQL.

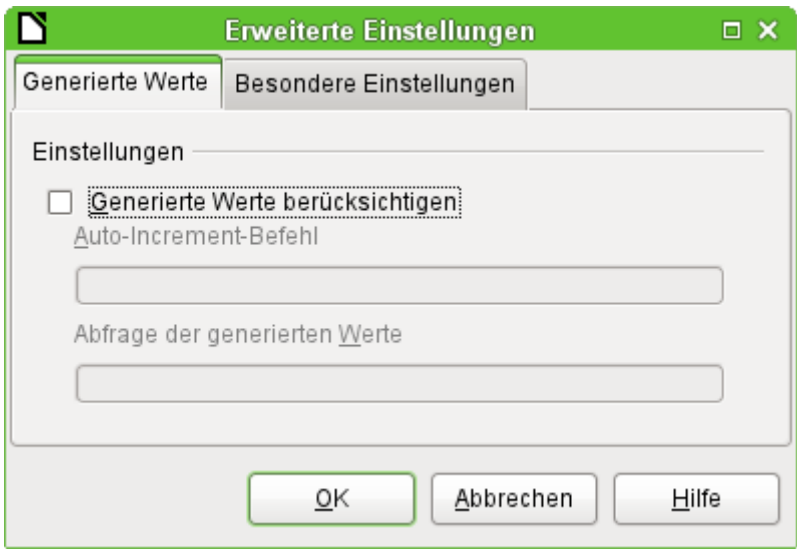



Gibt es Probleme mit dem Zeichensatz? Diese Probleme können eventuell in dem zweiten Reiter der Datenbankeigenschaften gelöst werden.

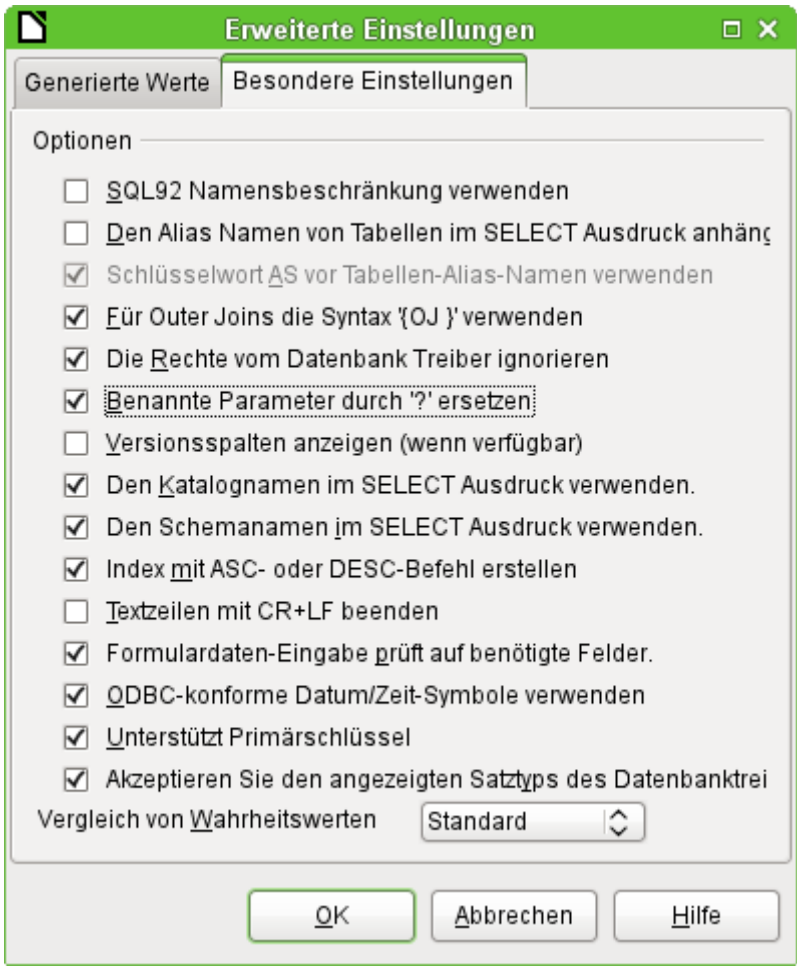
Auch eine zusätzliche spezielle Einstellung des Treibers ist möglich, wenn eventuell ein Parameter ausgeführt werden soll, der zur Zeit nicht bereits in der odbc.ini steht.




Wird die Verbindungsart ausgewählt, so kann der gesamte Kontakt zur Datenquelle geändert werden. Die darauf folgenden Schritte sind gleich denen des Datenbankassistenten ab Schritt 2. So kann z.B. von einer ODBC- auf eine JDBC- oder gar eine direkte Verbindung über die internen Treiber von LO gewechselt werden. Dies ist vor allem dann sinnvoll, wenn zuerst einmal ausgetestet werden soll, welche Verbindungsart zu dem eigenen Projekt am besten passt.




Je nach Datenbanksystem gibt es unterschiedliche Befehle um automatisch hochzählende Werte zu erzeugen. Soll so etwas erreicht werden, was der Treiber zur zeit nicht direkt ermöglicht, dann ist an dieser Stelle Handarbeit angesagt. Dabei ist sowohl ein Befehl für das Erstellen eines Auto-Wert-Feldes als auch ein Befehl zur Abfrage des zuletzt erstellten Wertes notwendig.




Die über **Bearbeiten → Datenbank → Erweitere Einstellungen** erreichbaren besonderen Einstellungen sind beeinflussen das Zusammenspiel von externer Datenbank und Base auf die unterschiedliche Art und Weise. Manche Einstellungen sind bereits ausgegraut, da sie für die zugrundeliegende Datenbank nicht änderbar sind. In dem obigen Beispiel wurde **Benannte Parameter**

**durch '?' ersetzen** ausgewählt. Es hatte sich gezeigt, dass sonst bei PostgreSQL die Weitergabe von Werten von einem Hauptformular zum Unterformular nicht funktionierte. Erst mit dieser Einstellung funktionierten die weitergehenden Formularkonstruktionen aus dem Kapitel «Formulare» dieses Handbuches korrekt.