

LibreOffice
The Document Foundation

Base Handbook

Appendix

Copyright

This document is Copyright © 2013 by its contributors as listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), version 3 or later, or the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later.

All trademarks within this guide belong to their legitimate owners.

Contributors

Jochen Schiffers
Hazel Russman

Robert Großkopf

Jost Lange

Feedback

Please direct any comments or suggestions about this document to:
documentation@global.libreoffice.org.

Caution



Everything you send to a mailing list, including your email address and any other personal information that is written in the mail, is publicly archived and cannot be deleted.

Acknowledgments

This chapter is based on an original German document and was translated by Hazel Russman.

Publication date and software version

Published 16 May 2013. Based on LibreOffice 3.5.

Note for Mac users

Some keystrokes and menu items are different on a Mac from those used in Windows and Linux. The table below gives some common substitutions for the instructions in this chapter. For a more detailed list, see the application Help.

<i>Windows or Linux</i>	<i>Mac equivalent</i>	<i>Effect</i>
Tools > Options menu selection	LibreOffice > Preferences	Access setup options
<i>Right-click</i>	<i>Control+click</i>	Open a context menu
<i>Ctrl (Control)</i>	<i>⌘ (Command)</i>	Used with other keys
<i>F5</i>	<i>Shift+⌘+F5</i>	Open the Navigator
<i>F11</i>	<i>⌘+T</i>	Open the Styles and Formatting window

Contents

Copyright	2
Contributors.....	2
Feedback.....	2
Acknowledgments.....	2
Publication date and software version.....	2
Note for Mac users	2
Barcodes	4
Data types for the table editor	4
Integers.....	4
Floating-point numbers.....	4
Text.....	5
Time.....	5
Other.....	5
Built-in functions and stored procedures	6
Numeric.....	6
Text.....	7
Date/Time.....	8
Database connection.....	9
System.....	9
Information tables for HSQLDB	10
Database repair for *.odb files	11
Connecting a database to an external HSQLDB	13
Changing the database connection to external HSQLDB	15
Changing the database connection for multi-user access.....	15
Auto-incrementing values with external HSQLDB.....	17

Barcodes

To be able to use the barcode print function, the font ean13 . t t f must be installed. This font is freely available.

EAN13 barcodes can be created using ean13 . t t f as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number	Upper case, A=0 B=1 etc.						*	Lower case, a=0 b=1 etc.						+

See also the query **Barcode_EAN13_ttf_command** in the example database Media_without_Macros.

Data types for the table editor

Integers				
<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Range</i>	<i>Storage space</i>
Tiny Integer	TINYINT	TINYINT	$2^8 = 256$ - 128 bis + 127	1 Byte
Small Integer	SMALLINT	SMALLINT	$2^{16} = 65536$ - 32768 bis + 32767	2 Byte
Integer	INTEGER	INTEGER INT	$2^{32} = 4294967296$ - 2147483648 bis + 2147483647	4 Byte
BigInt	BIGINT	BIGINT	2^{64}	8 Byte
Floating-point numbers				
<i>Type</i>	<i>Option</i>	<i>HSQldb</i>	<i>Range</i>	<i>Storage space</i>
Decimal	DECIMAL	DECIMAL	Unlimited, up to 50 places in the GUI, fixed decimal point, perfect accuracy	variable
Number	NUMERIC	NUMERIC	Unlimited, up to 50 places in the GUI, fixed decimal point, perfect accuracy	variable
Float	FLOAT	(DOUBLE used instead)		
Real	REAL	REAL		
Double	DOUBLE	DOUBLE [PRECISION] FLOAT	Adjustable, not exact, 15 decimal places maximum	8 Byte

Text				
<i>Type</i>	<i>Option</i>	<i>HSQLDB</i>	<i>Range</i>	<i>Storage space</i>
Text	VARCHAR	VARCHAR	Adjustable	variable
Text	VARCHAR_IGNORECASE	VARCHAR_IGNORECASE	Adjustable, range affects sorting	variable
Text (fix)	CHAR	CHAR CHARACTER	Adjustable, rest of actual text replaced with spaces	fixed
Memo	LONGVARCHAR	LONGVARCHAR		variable
Time				
<i>Type</i>	<i>Option</i>	<i>HSQLDB</i>	<i>Range</i>	<i>Storage space</i>
Date	DATE	DATE		4 Byte
Time	TIME	TIME		4 Byte
Date/Time	TIMESTAMP	TIMESTAMP DATETIME	Adjustable (0.6 – 6 means with milliseconds)	8 Byte
Other				
<i>Type</i>	<i>Option</i>	<i>HSQLDB</i>	<i>Range</i>	<i>Storage space</i>
Yes/No	BOOLEAN	BOOLEAN BIT		
Binaryfield (fix)	BINARY	BINARY	Like Integer	fixed
Binary field	VARBINARY	VARBINARY	Like Integer	variable
Image	LONGVARBINARY	LONGVARBINARY	Like Integer	variable, intended for larger images
OTHER	OTHER	OTHER OBJECT		

Built-in functions and stored procedures

Numeric	
As we are dealing here with floating point numbers, be sure to take care with the settings of the fields in queries. Mostly the display of decimal places is restricted, so that in some cases there may be unexpected results. For example, column 1 might show 0.00 but actually contain 0.001, and column 2, 1000. If column 3 is set to show Column 1 * Column 2, it would actually show 1.	
ABS(d)	Returns the absolute value of a number, removing a minus sign where necessary.
ACOS(d)	Returns the arc-cosine.
ASIN(d)	Returns the arc-sine.
ATAN(d)	Returns the arc-tangent.
ATAN2(a,b)	Returns the arc-tangent using coordinates. a is the value of the x-axis, b the value of the y-axis
BITAND(a,b)	Both the binary form of a and the binary form of b must have 1 at the same position to yield 1 in the result. BITAND(3,5) yields 1; 0011 AND 0101 = 0001
BITOR(a,b)	Either the binary form of a or the binary form of b must have 1 at the same position to yield 1 in the result. BITAND(3,5) yields 7; 0011 OR 0101 = 0111
CEILING(d)	Returns the smallest whole number that is not smaller than d.
COS(d)	Returns the cosine.
COT(d)	Returns the cotangent.
DEGREES(d)	Converts radians to degrees.
EXP(d)	Returns e^d (e: (2.718...)).
FLOOR(d)	Returns the largest whole number that is not greater than d.
LOG(d)	Returns the natural logarithm to base e.
LOG10(d)	Returns the logarithm to base 10.
MOD(a,b)	Returns the remainder as a whole number, in the division of 2 whole numbers. MOD(11,3) ergibt 2, weil $3*3+2=11$
PI()	Returns π (3.1415...).
POWER(a,b)	a^b , POWER(2,3) = 8, since $2^3 = 8$
RADIANS(d)	Converts degrees to radians.
RAND()	Returns a random number greater than or equal to 0.0 and less than 1.0.

ROUND(a,b)	Rounds a to b decimal places.
ROUNDMAGIC(d)	Solves rounding problems, that arise from using floating point numbers. 3.11-3.1-0.01 is not exactly 0, but is shown as 0 in the GUI. ROUNDMAGIC makes it an actual zero value.
SIGN(d)	Returns -1, if d is less than 0, 0 if d==0 and 1 if d is greater than 0.
SIN(A)	Returns the sine of an angle in radians.
SQRT(d)	Returns the square root.
TAN(A)	Returns the tangent of an angle in radians.
TRUNCATE(a,b)	Truncates a to b decimal places. TRUNCATE(2.37456,2) = 2.37
Text	
ASCII(s)	Returns the ASCII code of the first letter of the string.
BIT_LENGTH(str)	Returns the length of the text string str in bits.
CHAR(c)	Returns the letter corresponding to the ASCII code c.
CHAR_LENGTH(str)	Returns the length of the text in characters.
CONCAT(str1,str2)	Concatenates str1 and str2.
'str1' 'str2' 'str3' or 'str1'+ 'str2'+ 'str3'	Concatenates str1 + str2 + str3, simpler alternative to CONCAT.
DIFFERENCE(s1,s2)	Returns the sound difference between s1 and s2. Only a whole number is output. 0 means they sound the same. So 'for' and 'four' yield 0, 'king' and 'wing' yield 1, 'see' and 'sea' yield 0.
HEXTORAW(s1)	Translates hexadecimal code to other characters.
INSERT(s,start,len,s2)	Returns a text string, with part of the text replaced. Beginning with start, a length len is cut out of the text s and replaced by the text s2. INSERT(Bundesbahn, 3, 4, mmel) converts Bundesbahn into Bummelbahn, where the length of the inserted text can be greater than that of the deleted text without causing any problems. So INSERT(Bundesbahn, 3, 5, s und B) yields 'Bus und Bahn'.
LCASE(s)	Converts a string to lower case.
LEFT(s,count)	Returns the first count characters from the beginning of the text s.
LENGTH(s)	Returns the length of text in characters.
LOCATE(search,s,[start])	Returns the first match for the term search in the text s. The match is given as an offset number: (1=left, 0=not found) Setting a starting point within the text string is optional.
LTRIM(s)	Removes leading spaces and non-printing characters from the beginning of a text string.

OCTET_LENGTH(str)	Returns the length of a text string in bytes. This corresponds to twice the length in characters.
RAWTOHEX(s1)	Converts to hexadecimals, reverse of HEXTORAW().
REPEAT(s,count)	Repeats the text string s count times.
REPLACE(s,replace,s2)	Replaces all existing occurrences of replace in the text string s by the text s2.
RIGHT(s,count)	Opposite of LEFT; returns the last count characters at the end of a text string.
RTRIM(s)	Removes all spaces and non-printing characters from the end of a text string.
SOUNDEX(s)	Returns a 4-character code, corresponding to the sound of s – matches the function DIFFERENCE().
SPACE(count)	Returns count spaces.
SUBSTR(s,start[,len])	Abbreviation for SUBSTRING.
SUBSTRING(s,start[,len])	Returns the text s from the start position (1=left). If length is left out, the whole string is returned.
UCASE(s)	Converts a string to upper case.
LOWER(s)	As LCASE(s)
UPPER(s)	As UCASE(s)
Date/Time	
CURDATE()	Returns the current date.
CURTIME()	Returns the current time.
DATEDIFF(string, datetime1, datetime2)	Date difference between two dates- compares date/time values. The entry in string determines the units in which the difference is returned: ms=millisecond, ss=second, mi=minute, hh=hour, dd=day, mm=month, yy = year. Both the long and the short forms can be used for string.
DAY(date)	Returns the day of the month (1-31).
DAYNAME(date)	Returns the English name of the day.
DAYOFMONTH(date)	Returns the day of the month (1-31). Synonym for DAY()
DAYOFWEEK(date)	Returns the weekday as a number (1 represents Sunday).
DAYOFYEAR(date)	Returns the day of the year (1-366).
HOUR(time)	Returns the hour (0-23).
MINUTE(time)	Returns the minute (0-59).

MONTH(date)	Returns the month (1-12).
MONTHNAME(date)	Returns the English name of the month.
NOW()	Returns the current date and the current time together as a timestamp. Alternatively CURRENT_TIMESTAMP can be used.
QUARTER(date)	Returns the quarter of the year (1-4).
SECOND(time)	Returns the seconds part of the time (0-59).
WEEK(date)	Returns the week of the year (1-53).
YEAR(date)	Returns the year part of a date entry.
CURRENT_DATE	Synonym for CURDATE(), SQL-Standard, <i>Base gives the message: Access is denied.</i>
CURRENT_TIME	Synonym for CURTIME(), SQL-Standard.
CURRENT_TIMESTAMP	Synonym for NOW(), SQL-Standard.

Database connection

Except for IDENTITY(), which has no meaning in Base, all these can be carried out using **Direct SQL Command**.

DATABASE()	Returns the name of the database to which this connection belongs.
USER()	Returns the username of this connection.
CURRENT_USER	SQL standard function, synonym for USER().
IDENTITY()	Returns the last value for an autovalue field, which was created in the current connection. This is used in macro coding to transfer a primary key in one table to become a foreign key for another table.

System

IFNULL(exp,value)	If exp is NULL, value is returned, otherwise exp. Alternatively as an extension COALESCE() can be used. Exp and value must have the same data type.
CASEWHEN(exp,v1,v2)	If exp is true, v1 is returned, otherwise v2. Alternatively CASE WHEN can be used. CASE WHEN works better with the GUI.
CONVERT(term,type)	Converts term into another data type.
CAST(term AS type)	Synonym for CONVERT().
COALESCE(expr1,expr2, expr3,...)	If expr1 is not NULL, returns expr1, otherwise expr2 is checked, then expr3 and so on.
NULLIF(v1,v2)	If v1 is equal to v2, NULL is returned, otherwise v1.
CASE v1 WHEN v2 THEN v3 [ELSE v4] END	If v1 is equal to v2, v3 is returned. Otherwise v4 is returned or NULL, if there is no ELSE clause.

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.
EXTRACT ({YEAR MONTH DAY HOUR MINUTE SECOND} FROM <date or time>)	Can replace many of the date and time functions. Returns the year, the month, the day, etc. from a date or date/time value.
POSITION(<string expression> IN <string expression>)	If the first string is contained in the second one, the offset of the first string is given, otherwise 0 is returned.
SUBSTRING(<string expression> FROM <numeric expression> [FOR <numeric expression>])	Yields part of a text string from the position specified in FROM, optionally up to the length given in FOR.
TRIM([{LEADING TRAILING BOTH}] FROM <string expression>)	Non-printing special characters and spaces are removed.

Information tables for HSQLDB

Inside a database, information on all table properties and their connections to one another are stored in the *INFORMATION_SCHEMA* area. This information allows Base macros to be created that require very few arguments for their procedures. An application is to be found in the example database in the *Maintenance* module—the *Table_purge* procedure for the control of dialogs.

In a query, individual pieces of information and all the fields that belong can be provided in the following way:

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

In contrast to a normal table, it is necessary here to use *INFORMATION_SCHEMA* as a prefix to the appropriate name from the following list:

```
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

Database repair for *.odb files

Regular backing up of data should be standard practice when using a PC. Backup copies are the simplest way to return to an even halfway current state for your data. However, in practice this is often lacking.

In the case of sudden PC crashes, it can happen that open databases (internal HSQLDB databases) can no longer be opened in LibreOffice. Instead, when you attempt to open the database, you are asked for a filter corresponding to the format.

The problem here is that part of the data in an open database is contained in working memory and is only temporarily copied to intermediate storage. Only when the file is closed is the whole database written back into the file and repacked.

To get access again to your data, you may find the following procedure helpful:

1. Create a copy of your database for the steps that follow.
2. Try to open the copy with an archiving program. In the case of *.odb files, we are dealing with a compressed format, a Zip archive. If the file cannot be opened directly, try renaming it from *.odb to *.zip. If that does not open it, your database is past saving.
3. The following folders will always be seen after opening a database file in an archiving program:



4. The database file must be decompressed. The most important information, as far as the data is concerned, are in the subfolder database in the files data and script.
5. It may be necessary to look at the script file and test it for contradictions. This step can, however, be left for the testing stage. The script file contains above all the description of the table structure.
6. Create a new empty database file and open this file with the archiving program.
7. Replace the files data and script in the new database file with the files unpacked in step 4.
8. Close the archiving program. If it was necessary to rename the file to *.zip before opening it in the archiving program (this depends on your operating system), now rename it again to *.odb.
9. Open the database file in LibreOffice. You should be able to access your tables again.
10. How far your queries, forms, and reports can be recovered in a similar way must be the subject of further testing.

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

If, as described in the following pages, you are using external HSQLDB, there may be a further problem with the *.odb file connected with some LibreOffice versions. If external HSQLDB is used, the safest way is through the hsqldb.jar-archive, which is supplied with LibreOffice. If a different archive is used, it can lead to the internal database suddenly becoming inaccessible. This is because LibreOffice 3.x has difficulty distinguishing between internal and external HSQLDB and produces warnings of an incompatibility between versions.

If internal databases can no longer be opened, you must use as your external database the supplied hsqldb.jar file. In addition, you must extract from the *.odb file the database folder. The properties file in this case has an entry that leads to the above error in LibreOffice:

```
version=1.8.1 on line 11
```

This line should be changed to:

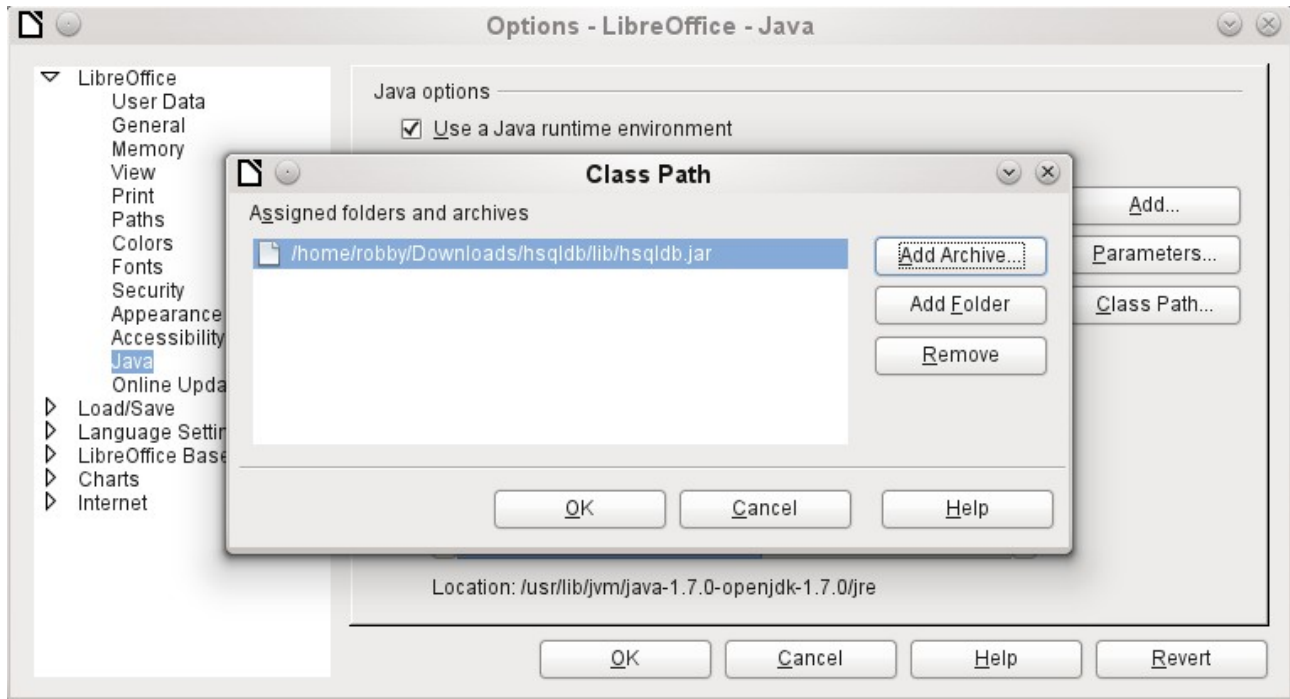
```
version=1.8.0
```

Afterwards the database folder is put back into the *.odb package and the database can once more be opened in LibreOffice.

Connecting a database to an external HSQLDB

Internal HSQLDB is indistinguishable from the external variant. If, as in the following description, the initial access to the database is from the outside, no server function is necessary. You just need the archive program which is supplied with LibreOffice. You will find it on the path under **/program/classes/hsqldb.jar**. The use of this archive is the safest solution, as you then get no version problems.

External HSQLDB is freely available for download at <http://hsqldb.org/>. When the database is installed, the following steps must be performed in LibreOffice:



If the database driver does not lie on the Java-Runtime path, it must be entered as a Class Path under **Tools > Options > Java**.

The connection to the external database uses JDBC. The database file should be stored in a particular directory. This directory can be freely chosen. In the following example it is in the home folder. The rest of the directory path and the name of the database are not given here.

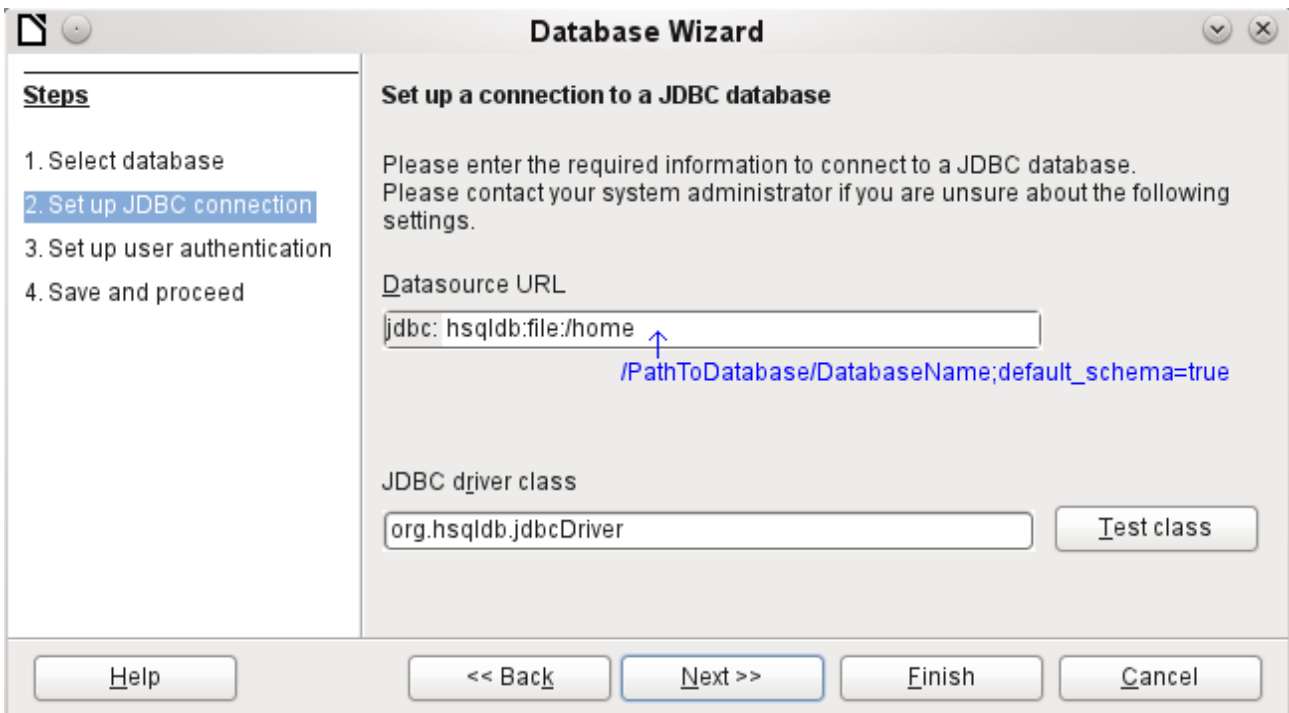
It is important, if data in the database are to be written using the GUI, that next to the database name the words **" ; default_schema=true "** are written.

So:

```
jdbc:hsqldb:file:/home/PathToDatabase/Databasename;default_schema=true
```

In the folder you will find the files:

```
Databasename.backup  
Databasename.data  
Databasename.properties  
Databasename.script  
Databasename.log
```



The next step is to give the default user, if nothing in the HSQLDB configuration is to be changed:



This creates the connection and the database becomes accessible.

Caution



If an external database is edited with a version of HSQLDB 2.x, it can no longer be converted into an internal database under LibreOffice. This is because of additional functions which are not present in version 1.8.x. This terminates the invocation in the case of version 1.8.x while the script file of the database is being read in.

In the same way an external database which has once been edited with a version of the second series cannot afterwards be edited with version 1.8.x, which is compatible with LibreOffice.

Changing the database connection to external HSQLDB

Internal HSQL databases have the disadvantage that data storage involves a compressed archive. Only on compression are all the data finally written. This can more easily lead to data loss than when working with an external database. The following section shows the steps necessary to successfully change an existing database from an *.odb archive to an external version in HSQL.

From a copy of the existing database, extract the database directory. Copy the contents into an arbitrary directory as described above. Add the database name to the resultant filenames:

```
Databasename.backup
Databasename.data
Databasename.properties
Databasename.script
Databasename.log
```

Now the content .xml file must be extracted from the *.odb archive. Use any simple text editor to find the following lines:

```
<db:connection-data><db:connection-resource
xlink:href="sdbc:embedded:hsqldb"/><db:login db:is-password-
required="false"/></db:connection-data><db:driver-settings/>
```

These lines must be replaced with a connection to an external database, in this case a connection to a database with the name Union, in the hsqldb_data directory.

```
<db:connection-data><db:connection-resource
xlink:href="jdbc:hsqldb:file:/home/robby/documents/databases/hsqldb_da
ta/Union;default_schema=true"/><db:login db:user-name="sa" db:is-
password-required="false"/></db:connection-data><db:driver-settings
db:java-driver-class="org.hsqldb.jdbcDriver"/>
```

If, as described above, the basic configuration of HSQLDB was not damaged, the username and the optional password must also agree.

After changing the code, content .xml must be put back into the *.odb archive. The database directory in the archive is now surplus to requirements. The data will in the future be accessed through the external database.

Changing the database connection for multi-user access

For multi-user access, HSQLDB must be made available over a server. How the installation of the server is carried out varies depending on your operating system. For OpenSuSE, it is only necessary to download the appropriate package and to start the server centrally using YAST (runlevel setting). Users of other operating systems and other Linux distributions can likely find suitable advice on the Internet.

In the home directory on the server (in SuSE, **/var/lib/hsqldb**), you will find, among other things, a directory called data, in which the database is to be filed, and a file called server.properties, which controls the access to the databases in this directory.

The following lines reproduce the complete contents of this file on my computer. It controls access to two databases, namely the original default database (which can be used as a new database) and the database that was extracted from the *.odb file.

```
# Hsqldb Server cfg file.
# See the Advanced Topics chapter of the Hsqldb User Guide.

server.database.0    file:data/db0
server.dbname.0     firstdb
server.urlid.0      db0-url
```



```

server.database.1    file:data/union
server.dbname.1     union
server.urlid.1      union-url

server.silent       true
server.trace        false

server.port         9001
server.no_system_exit true

```

The database 0 is addressed with the name `firstdb`, although the individual files in the data directory begin with `db0`. I added my own database as Database 1. Here the database name and file begin identically.

The two databases are addressed in the following way:

```

jdbc:hsqldb:hsqldb://localhost/firstdb;default_schema=true
username sa
password
jdbc:hsqldb:hsqldb://localhost/union;default_schema=true
username sa
password

```

The suffix `;default_schema=true` to the URL, which is necessary for write access using the graphical user interface of LibreOffice, is permanently included.

If you actually need to work on the server, you will want to consider if the database needs to be password-protected for security reasons.

Now you can connect to the server using LibreOffice.

With this access data, the server can be loaded on its own computer. On a network with other computers, you must give either the host name or the IP address to the server, which in this case is running on my computer.

Example: My computer has the IP 192.168.0.20 and is known on the network by the name `lin_serv`. Now suppose there is another computer to be entered for connection to the database:

```

jdbc:hsqldb:hsqldb://192.168.0.20/union;default_schema=true

```

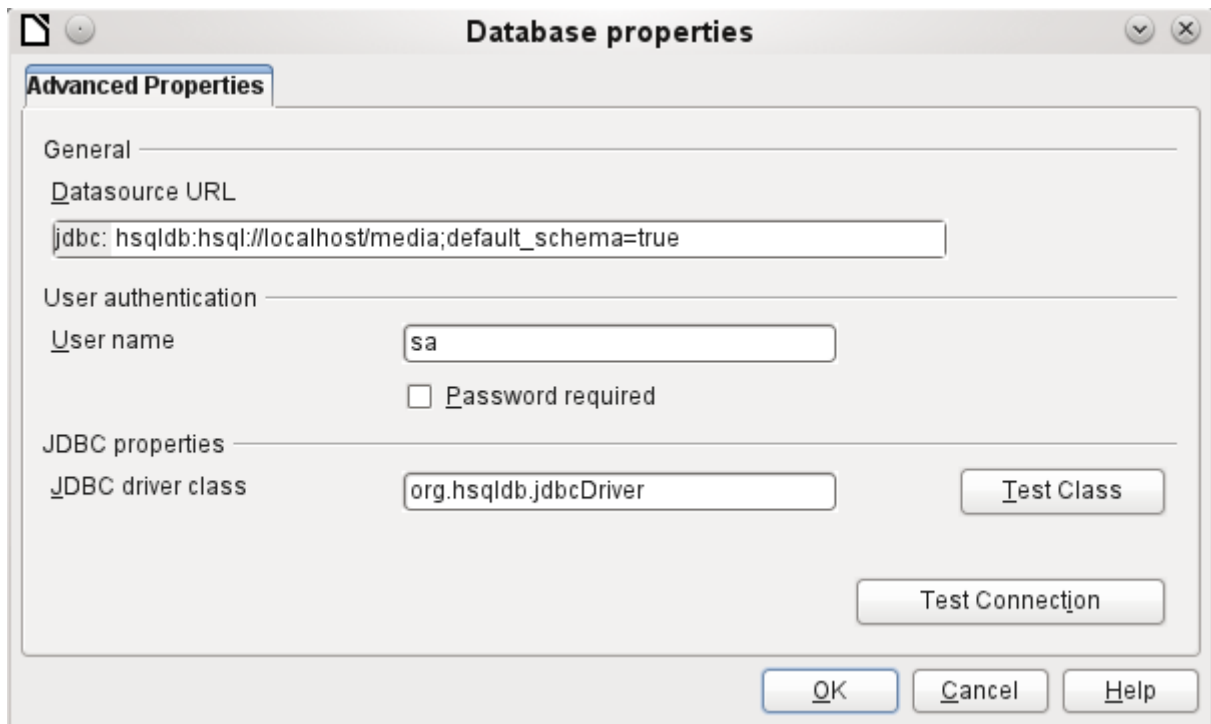
or:

```

jdbc:hsqldb:hsqldb://lin_serv/union;default_schema=true

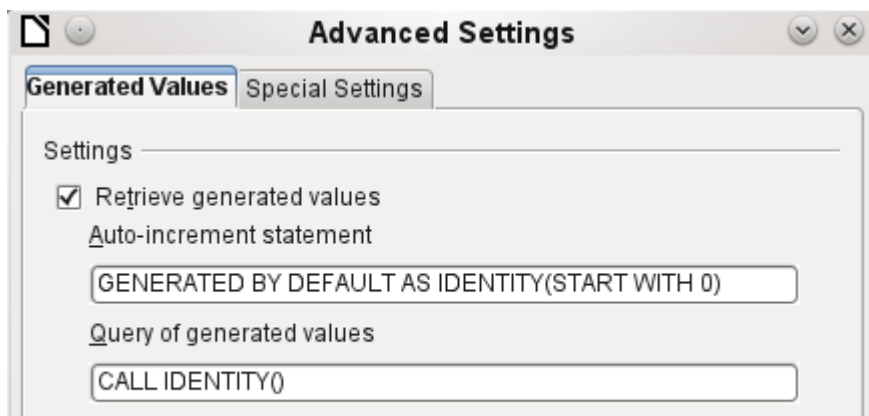
```

The database is now connected and we can write into it. Quickly, however, an additional problem appears. The previously automatically generated values are suddenly no longer incremented. For this purpose we need an additional setting.



Auto-incrementing values with external HSQLDB

To use autovalues, different procedures for table configuration are needed according to the version of LibreOffice. Common to all of them is the following entry under **Edit > Database > Advance settings**:



Adding **GENERATED BY DEFAULT AS IDENTITY(START WITH 0)** causes the function of the automatically incrementing values for the primary key to be set. The GUI in LibreOffice takes up this command, but unfortunately prefaces the statement with **NOT NULL**, so that the command sequence for HSQLDB is not readable. Here you must ensure that HSQLDB gets sent the above command so that the corresponding field will contain the primary key.

Note

In LibreOffice the entry of autovalues using the GUI is not possible for this reason. Users of these versions should first create a table with a primary key field that does *not* auto-increment and then enter directly using **Tools > SQL**:

```
ALTER TABLE "Table_name" ALTER COLUMN "ID" INT GENERATED BY
DEFAULT AS IDENTITY(START WITH 0)
```

This assumes that the primary key field has the name ID.