Database in MatDeck

MatDeck has incorporated a range of functions and tools that can be used for simple and fast data exchange with SQLite database. You can store and read a large amount of data, combine this data with GUI elements to create very useful applications and forms, save results of your calculations in SQLite db files and share them, ...

We can divide database functions in two group:

- Functions for database data exchange
- Graphical objects for database data exchange

Functions for database data exchange are:

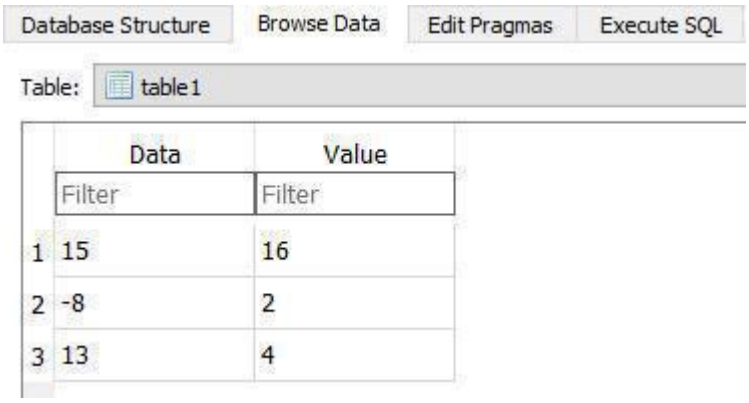
Database read - With this function you can read data stored in SQLite db documents and place them in MatDeck variable, ready for further use; Function database read syntax is:

```
database read("Document name.db" , "Table name")
```

Document name, path is relative to mdd document directory

Exact name table in database

For example:



Browse Data from 'DB Browser for SQLite'

```
dbn := user dir( ) + "/Flexitek/" + "Example data.db"
file copy("Example data.db" , user dir( ) + "/Flexitek")
a := database read(dbn , "table1")
```

Absolute file path is placed in variable dbn

With file copy function we have downloaded file "Example data.db" and will continue to read data from it's local copy

a =

Data	Value
15	16
-8	2
13	4

database write - with this function you can write data from a MatDeck document to database document; Function database write syntax is:

```
database write("Document name.db" , "Table name" , Value to write)
```

Document name, use Ctrl + Pointer to open Menu

Exact name of table in database

Value to write in database table

For example, if data is stored in variable b:

$$b := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

```
database write(dbn , "table2" , b)
```

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: table2

	id	column0	column1	column2
	Filter	Filter	Filter	Filter
1	1	1	2	3
2	2	4	5	6

As a last argument you can place any numerical value, vector, matrix, table or variable. If a inserted table name not exist in database, the table with that name will be created. In example above a database table with name 'table2' didn't existed until we called the database write function.

If we place a MatDeck table as a last argument of the database write function, names of columns in the database table will be the same as names of columns of the MatDeck table.

c :=

abc	cde
1	2
3	4

```
database write(dbn , "table3" , c)
```

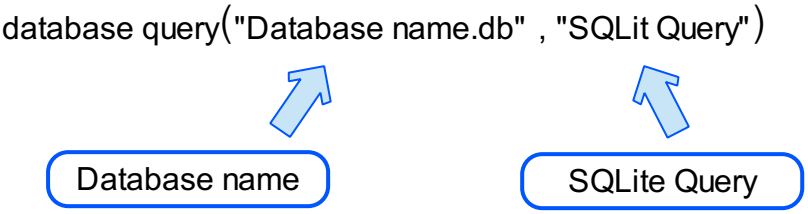
Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: table3

	id	abc	cde
	Filter	Filter	Filter
1	1	1	2
2	2	3	4

As you have notice, when we write data to a database a column with name 'id' is created in front of your data. This column is a primary key in this database table.

Database query - With this function you can execute queries in a selected database.You have to write queries as a string in SQLite query language, which is basically the standard SQL language but omits some features while at the same time adding a few features of its own.
Function database query syntax is:



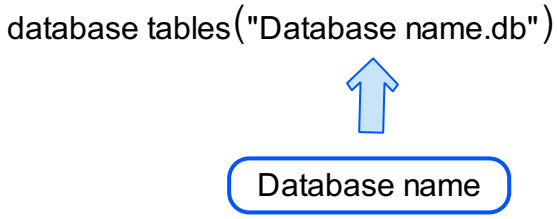
For example:

```
d := database query(dbn , "SELECT abc FROM table3 ORDER BY abc DESC")
```

d =

abc
3
1

database tables - use this function when you want to preview names of all tables in a specified database



For example:

```
e := database tables(dbn)
```

e =

```
["table1"
"table2"
"table3"]
```

database table columns - use this function when you want to preview names of all columns in a specified database table

```
database table columns("Database name.db" , "Table name")
```

Database name

Exact name of table in database

For example:

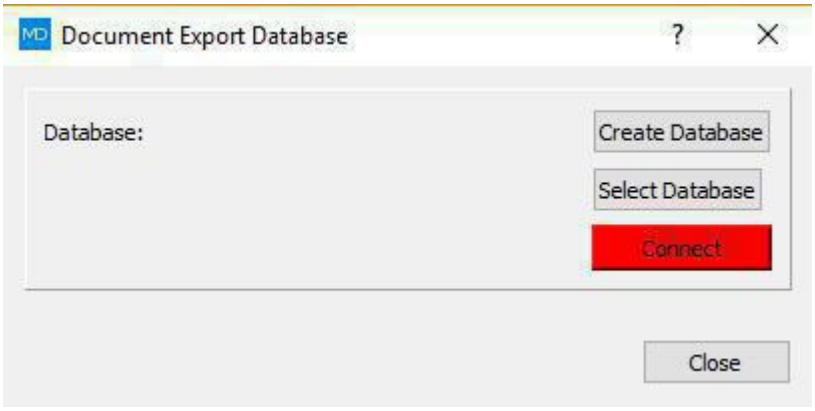
```
f:= database table columns(dbn , "table3")

f = ["id"  "abc"  "cde"]
```

```
file delete(dbn)
```

delete downloaded file

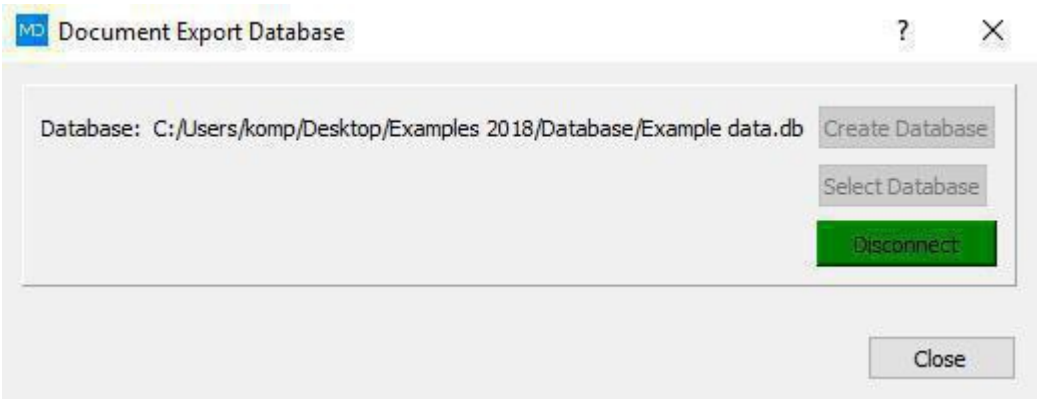
Graphical objects for database data manipulations are:

Database Manager - To create this object press the DB Manager icon from Data tab and the following window will open

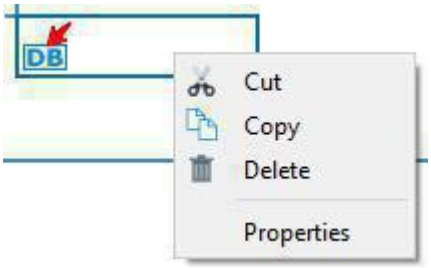


From this window you can choose to 'Create Database' or to connect to an existing database with option 'Select Database'.

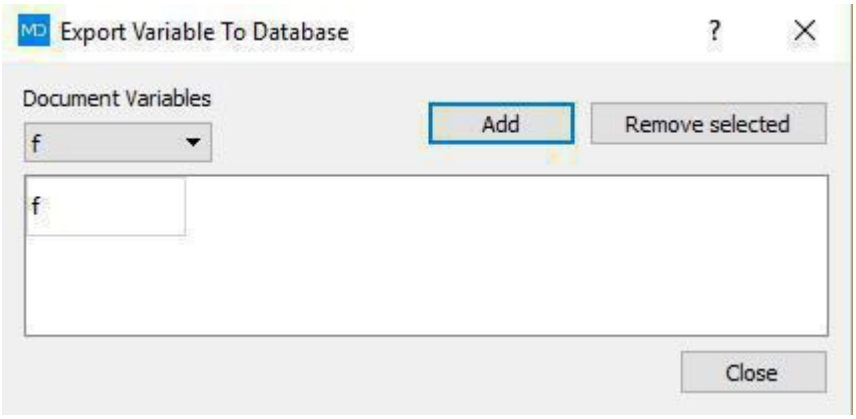
When you establish connection to a database the window will show the path to the database file and its name. If you press Connect button when database file is selected, button will go green and changed to Disconnect as shown on picture bellow.



Database Export - To create this object press the Export icon from Data tab (Database group) and click on the position on the canvas where you would like to place it. Pressing the right mouse button opens the content menu from which you should choose the Properties option.



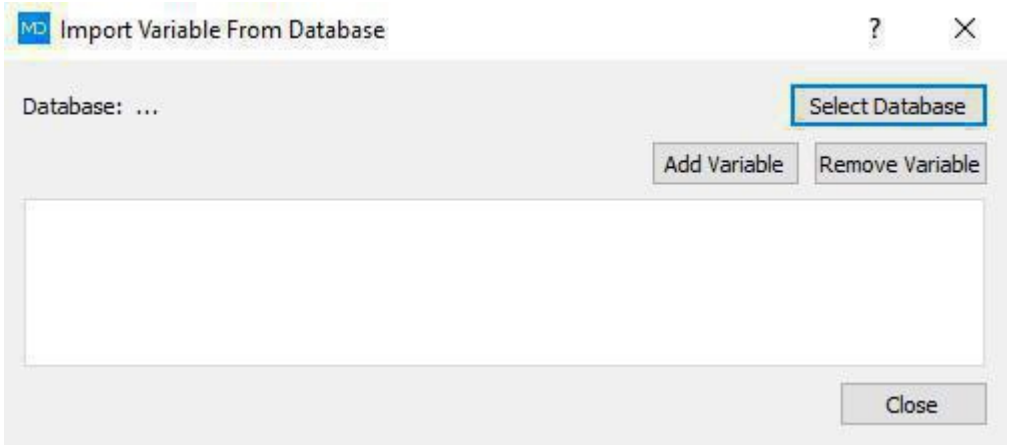
It will open new window, with combo box with the current document variables. You have to choose a variable which data will be exported to the database and press Add button. Data will be exported to a database table with the same name as the name of variable we are exporting, 'f' in this case.



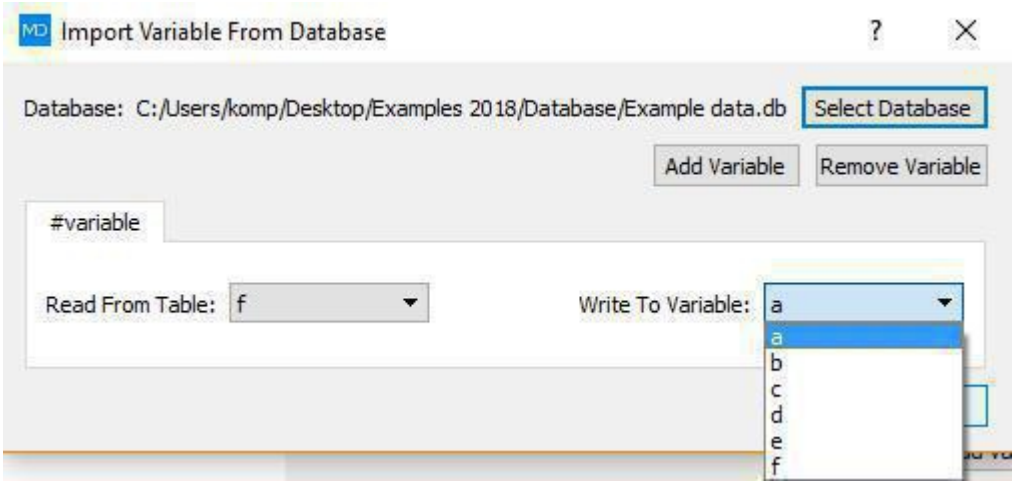
To remove this variable from export, select it and press 'Remove selected'.

Database Import - To create this object press the Import icon from Data tab (Database group) and click on the position on the canvas where you would like to place it. Pressing the right mouse button opens the content menu from which you should choose the Properties option.

It will open new window, select a database from which you want to import the data. Press 'Add Variable' and window will fill out with new combo boxes. You have to choose a table from which you will import the data and a variable to store data in (as shown on picture bellow).



Empty Import Variable From Database window



Import Variable From Database window after connection to Database