shucan in-memory database GUI client

Manual

Author: Yin Yi

company: Beijing Shucan Technology Co., Ltd.

version: 0.0.3

time: 2021-06-12

数蚕科技

Abstract

This is a ShuCan in-memory database GUI client help manual produced by Beijing ShuCan Technology Co., Ltd. (hereinafter referred to as **SCT**). The purpose of this manual is to provide a brief GUI client product usage method and provide an accurate guide method.

This manual consists of products function introduction, system composition, file operation, data operation, Editor's Guide, FAQ and contact us are composed of seven parts.

Contents

1 Function Introduction

1.1 Introduction to function

This product is a part of SCT memory database; it is the GUI version of the client, which is an enhancement of the command line client.

This product organizes a collection of functions in an orderly manner through menus and windows, which is more convenient for database management and operation.

These functions include the visualization of data objects (tables, columns) to add, delete, modify, and check; import and export data objects; and exchange external data (binary and text data) of data objects. Remote management and backup of the database. And corresponding operation sql statement generation, execution and result display.

This product provides an intelligent sql editing environment, which can conveniently perform common editing operations in sql language and instant error reminders.

This product provides editor environment configuration functions for program developers. Friendly editing operations can be carried out for different environments.

1.2 Functional changes

On June 12, 2021, the Japanese product enhanced the editor function, added the client remote backup and restore function, and added the filtering of table data and various editing operations.

On May 24, 2021, the Japanese product officially completed the operation functions related to editing files. Solve the problem of some Chinese table names and column names.

On May 12, 2021, the initial version of the Japanese product is officially completed. This product is an improvement of the command-line client function, and aims to solve the unintuitive data object operation of the command client.

2 System composition

2.1 Software Composition

The sct_mdb_cli.exe of this product is the main program for execution, config is the configuration file, and the corresponding doc directory contains the user guide for this product. Other files are dependent on the operating environment. Do not move or change other files except the main program and configuration files.

2.2 Software Operation Requirements

This product runs on Microsoft© Windows 7 Service Pack 1 and above system platforms.

The installation program directory reserves at least 100M of hard disk space.

It is recommended to reserve at least 128M of memory when the program is running.

This product has a lot of interface components, and it is recommended that the screen resolution be at least 1920x1080.

2.3 Installation, startup and configuration

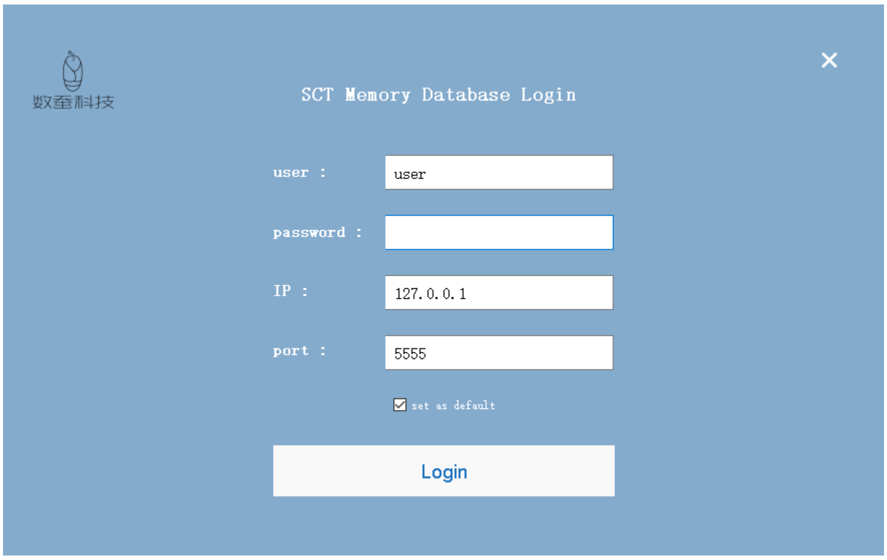
This product provides complete executable programs and help files.

When installing, directly use 7-Zip to keep the directory to decompress the official installation package.

You can run sct_mdb_cli.exe directly at startup. If the program starts abnormally, you may need to install the Microsoft Visual C++ (2019) runtime environment.

The configuration file of this program is a config file, which is a binary file, which retains your interface structure and personal preference setting information. You can keep this configuration file for future use, or use the default configuration.

After startup, the system enters the login window, please make sure that your server starts normally (if your server does not start normally, please refer to the server startup guide) and you have the appropriate server password access authority.



user : user

password :

IP : 127.0.0.1

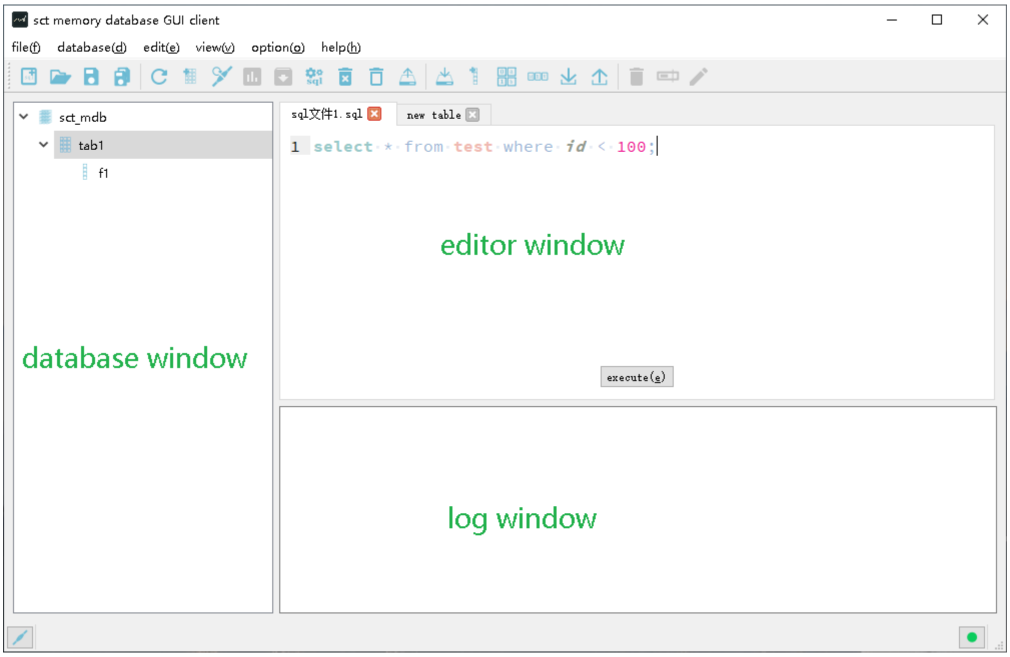
port : 5555

☑ set as default

Use your user name and password to log in to the server system, and check the option to automatically use the corresponding IP port and user name when starting up by default.

2.4 Interface Composition

The interface functions of this product include complete functional modules as shown in the figure below.



```
1 select * from test where id < 100;
```

Corresponding windows will be referred to as database, log, editor, and query results hereinafter.

The editor windows side by side can contain create table, edit table and column windows.

2.5 Interface adjustments

The **view** menu of the product can adjust various interfaces to form windows, menus and toolbars. You can customize the display of various windows according to your needs.

In the **view** menu, **icon** can adjust the size of the interface icons. Currently, three icon styles are available: **large**, **medium**, and **small**.

The corresponding window composition and icon information will be saved to the config file, and it will still take effect next time you start it.

The editor, query, and log windows can be dragged to adjust the size of each window.

2.6 Language adjustment

The **language** option in the **options** menu can be switched among the following languages

English
Chinese

2.7 Personal Preferences

preferences in the **options** menu provides preferences for the editor-related parameters. Including code color theme, automatic completion parameters, editing status display parameter settings.

2.8 Window Operation Guide

The database window is the database operation object, which is displayed as the database → table → column according to the data organization structure.

Selecting different operation objects can perform object operations through the right-click menu of the project, the database menu or the toolbar.

When operating the editor, you can use the corresponding editing menu content according to the editing content.

The log window displays the standard SQL statements executed by each operation, and outputs the execution time and running time. When the execution is successful, it is displayed in green, when the execution failed is displayed in red, it is displayed in green when the execution is successful, and the execution failure is displayed in red.

2.9 System Exit

This product uses the **exit** menu under the **file** menu to exit, or directly click the main window close button.

3 File Operation

3.1 File creation, opening

This product uses **new** under the **file** menu or double-click a new sql file in the blank space of the file editor. The default is to generate an unused file name for the created sql, and the newly created file must specify a file storage location.

This product uses **open** under the **file** menu to open a saved sql file. If the target file has been opened in the file editor, the corresponding editor window will be displayed.

If the file is opened or newly created successfully, it will be displayed in the corresponding editor window.

You can choose to open recently used files through the **open recent** menu. By default, 20 historical files are saved. You can use the **Clear List** submenu to clear your open records.

3.2 File save

This product uses **save** under the **file** menu to save the current editor sql file.

This product uses **save all** under the **file** menu to save all editor sql files.

This product uses **save as** under the **file** menu to save the current editor sql file.

3.3 File Close

This product uses **close** under the **file** menu to close the current editor sql file. When closing the file, the save operation will be performed by default, and then the corresponding editor window will be closed.

This product uses **close all** under the **file** menu to close all editor sql files. When closing the file, the save operation will be performed by default, and then all editor windows will be closed.

This product uses **close others** under the **file** menu to close the non-current editor sql file. When closing the file, the save operation will be performed by default, and then the non-current editor window will be closed.

3.4 File printing

This product uses **Print** under the **file** menu to print the current editor sql file. At present, the print settings are only carried out according to the editor page size.

3.5 Exit

This product uses the **exit** menu under the **file** menu to exit the system, save all the editor content when exiting, and save the state of the open file window. Note: The content of non-editor pages will not be saved when exiting.

4 Data operation

4.1 Description

The operation object of this product is the ShuCan memory data object. The basic object of the ShuCan memory database is the data table. Multiple data tables form a library.

The table uses columnar storage fields, so the data table structure object is a column (or called a field).

Each object provides routine addition, deletion, modification, and check operations.

4.2 Database Operation

This product provides database server for in-memory databases **reconnect**, **refresh**, **create table**, **load table**, **backup**, **restore** and **execute sql** file function.

To perform database operations, first select the root database node in the project window and click the right button or the corresponding button in the database menu or toolbar.

reconnect is used to determine whether the server is working normally. After the reconnection is successful, the status icon in the lower left corner of the system status bar is connected, otherwise it will display disconnected status.

refresh is used to display the node tree on the left. It is created by default, and the node status is not updated on the left side of the modified table structure. Use refresh to display the current server database structure status.

create table is used to create a new memory table. To create a new table, you need to specify the table name and create at least one field. Each field needs to specify the field name, field type, and whether it can be repeated. Use the +/- to the right of the table name to add or delete the currently selected field. After clicking Apply, the corresponding table is created, and the log window outputs the executed statement, result and time. For corresponding table names, field names and more specifications, please refer to online ShuCan sql specification.

The function of **load table** is to directly load a binary or sql file as a memory table. The binary table and sql file area are also faster and smaller in binary saving and loading, and the sql file has better portability and error correction. You can choose the file format according to your needs. The loading table can be loaded through the server or the client table file can be transmitted. The loading client reads all data at once, so it may not be suitable for particularly large data files. It is recommended to use server loading for large files. Load table can specify binary table file and sql table file. Binary table file does not contain table name information, so you need to specify the imported table name. The sql file is constructed using the create statement without specifying the table name. When using the server, make sure that the input table file exists on the server when saving. Otherwise, it may fail to load the data table. Either way to load the new table, please make sure that the corresponding table does not exist in the library, otherwise loading the table will cause a failure.

The function of **backup** is to back up the entire database as a single data file. The database backs up all tables as binary or sql files according to your choice. In the replacement mode, the existing file will be directly replaced. The corresponding server-side and client-side functions are similar to the loading table. It is recommended to use server-side loading for large files.

The function of **restore** is to restore the entire database from a single data file. The database restores the entire database from the original backup according to your choice. Note that restoring the database will restore all data tables to the target file state. Please pay attention to the data backup when restoring. The functions of the corresponding server and client are similar to those during backup. It is recommended to use server loading for large files.

The **execute sql** file directly executes the sql script file on the client. If there is an error in the execution of the corresponding file, the error location is given or the execution is successful.

4.3 Data table operation

Select the table node on the left to perform table operations on the database.

This product provides **delete table**, **empty table**, **view field**, **add field**,

view or modify data and **import of the entire table for the presentation operation** , **export**, **save** operations.

delete table is used to delete the table object of the specified node name from the in-memory database. After deleting, you need to refresh the node tree on the left to view the database structure.

clear table is used to quickly clear the memory data table, but still retain the table structure information.

view field structure displays the field structure information of the data table, including field name, field type and whether it is repeated.

add field is used to add new fields to the table, field type and SQL specification overview The type names are consistent. And set to the default value. It can be used on the right side of the table name in the add field window The button identified by the +/- sign adds or deletes a field. Each line of field information can be directly edited using a table. After clicking Apply, SQL is generated and the corresponding information of the executed statement is output in the log window.

view or modify data The table data display window, the default window is on the far left. The data view window outputs the first 1000 rows of data in the table by default. You can cancel the table output restriction and display all table data. At the same time, you can use filter conditions to filter data, and the filter conditions default to the content of where conditions. Use the where condition to filter content can be directly modified in the table, and after clicking Apply, the modified content of the corresponding table will be rewritten into the memory data table. Post-conditions can further filter and process the output data. For the corresponding post-conditions, please refer to post-processing flow. Note: After some post-conditions are processed, the data table content cannot be modified, and you may not be able to edit the corresponding output table content.

Some commonly used filter fields are added by default in the filter and post-processing drop-down boxes. You can also filter the corresponding table content by selecting the right-click menu **filter** or **add filter** for the table content.

The data window uses the field list to filter the displayed fields. Checking or canceling the corresponding table field will display or not display the corresponding field data. Multiple display formats can be used in the data window. The normalized display of all data is in the standard form of SCT' sql, which includes the sign bit display, integer value hexadecimal display, and suffix symbol display. Hexadecimal display shows that all integer values are hexadecimal format. When editing, they are grouped by 16 bits, and the groups are displayed with spaces. You can add or delete group spaces when editing. The space has no effect on the target value. Binary display shows that all integer values are binary, 4 digits are grouped when editing, and the groups are displayed with spaces. You can add or delete group spaces when editing. The space has no effect on the target value. Decimal display shows that all integer values are decimal. When editing, they are grouped with 3 digits, and the groupings are displayed with spaces. You can add or delete group spaces when editing. The space has no effect on the target value. After the modification, the data will be displayed in green by default in the immediate mode, and the immediate mode will be updated and modified immediately.

Find and replace Use the menu search or replace to search or replace the current table according to constraints. The found objects are displayed in yellow. By default, there is no range constraint, that is, the full table range. The row constraint only finds or replaces the current row, and the column constraint only finds or replaces the current column.

The search function supports case-sensitive search and whole word search options. Use the normal mode to query the target string in the full text without special processing. Use the regular expression mode to treat the target string as a regular expression, and the default regular expression uses the C++17 default regular expression parameter string to process. Use extended mode to target string only standard escape (C++17 standard) processing in the target string.

Use the **next** and **previous** buttons to find the next and previous matching strings at the current position. Find all displays all matching strings.

Use replace to replace the current string, and replace all to replace all.

The **import/export** of the table is used to import/export the csv or csvs file for exchange. For the csv or csvs file specification, please refer to import function instructions. Note that the server file path is used when importing the table, please make sure that the server has this file path. Select the field separator according to the content of the imported file. If the separator does not exist, you can directly enter the separator string in the drop-down box. The separator can be any number of non-end-of-line (\n) symbols.

The **save** function of the table is to directly save the memory database as a binary file, which can be used for fast data storage. The saved file is always a complete data table content. Please make sure that there is no file with the same table name on the server side when saving. Otherwise, it may fail to save the data.

4.4 Data field operation

Select the left field node to perform field operations on the database.

This product provides **delete characters**, **rename field**, and **modify field** functions for field operations.

delete field is used to delete the field object of the specified node name from the memory data table. After deleting, you need to refresh the node tree on the left to view the database table structure.

rename field can modify the target field name to the new field name, other information cannot be modified.

modify field is used to modify the field type and whether to repeat the parameters. The field name cannot be modified, and you can also add more field information through the modify field function. The button identified by the +/- sign adds or deletes a field. Each line of field information can be directly edited using a table. After clicking Apply, SQL is generated and the corresponding information of the executed statement is output in the log window.

5 Editor's Guide

5.1 sql specification

The sql code in this product should comply with SCT sql specification version 0.2.0. The basic requirement is that the editing code encoding defaults to the UTF-8 format, and the end of the line uses the \n format. For others, please refer to the above online reference.

5.2 Code style and theme

This product provides three styles of theme styles for different coders. They are **elegant**, **fresh** and **quiet**. You can go through the **options** menu Set the editor style and theme in **preferences**.

For different highlighted objects, you can select different objects and set the font and color style on the right to customize your own color theme. You can restore the default theme settings via **reset to default** on the right side of the theme.

The horizontal separation factor in the editor style setting indicates the ratio of the horizontal spacing between each character in the editor to the number of points in each font. The smaller the value, the smaller the horizontal distance between fonts.

The vertical separation factor in the editor style setting indicates the ratio of the line spacing in the editor to the number of points in each font. The smaller the value, the smaller the distance between rows.

The editor can display the space character by setting the ASCII 0x20 space character as a dot.

The color of the cursor can be set in the editing options of the editor.

Other style functions may not be implemented in the current version. Not recommended for use.

The editor line number display always starts from 1 to the last line of the editing content.

5.3 Code Editing

The SQL editor of this product supports commonly used editing functions. The default editing shortcut keys use the Windows style editing shortcut keys.

The shortcut key list is as follows:

`Ctrl+A`	`select all`
`Ctrl+Y`	`redo editing`
`Ctrl+Z`	`undo edit`
`Ctrl+C`	`copy selection`
`Ctrl+X`	`cut selection`
`Ctrl+V`	`paste`
`Ctrl+E`	`jump to error`
`Ctrl+W`	`select whole word`
`Ctrl+Left`	`go to the beginning of the word`
`Ctrl+Right`	`go to the end of the word`
`Home`	`go to the beginning of the line`
`Ctrl+F`	`find`
`Ctrl+H`	`replace`
`Delete`	`delete the selection or the next character`
`Backspace`	`delete selection or previous character`

Among them, Ctrl refers to the control key in the lower left corner of the keyboard, Left is the left arrow key (←), and Right is the right arrow key (→).

You can select multiple characters by holding down Shift and moving the left arrow key or clicking the mouse while holding down and dragging. Double-click to select a word by default, double-click again to select the entire line within 1 second, and double-click to select all within 1 second.

Selecting a word will show all the same words in yellow by default. Any edit action cancels the display content.

5.4 Find and Replace

Use the menu or the corresponding shortcut key to call up the search dialog box. The corresponding function works in the currently active editing window.

The search function supports case-sensitive search and whole word search options. Use the normal mode to query the target string in the full text without special processing. Use the regular expression mode to treat the target string as a regular expression, and the default regular expression uses the C++17 default regular expression parameter string to process. Use extended mode to target string only standard escape (C++17 standard) processing in the target string.

Use the **next** and **previous** buttons to find the next and previous matching strings at the current position. Find all displays all matching strings.

Use replace to replace the current string, and replace all to replace all.

5.5 Smart Tips

This product can provide intelligent completion prompts for edited sql. You can use **enable** or **disable** in the option section of the editor**autocompletion** in **preferences**. At present, as long as the input content has a prompt, the intelligent prompt is automatically performed, and only the sql language part and the content of the field table name are completed.

Auto-completion is an intelligent completion based on sql semantics, which uses the longest matching principle to parse the input sentence. The smart prompt does not access the server, so it is only for basic syntax and semantic analysis. Smart prompts cannot be used as an accurate basis for completion. For reference only when used.

When smart tips are available, a list of completion tips pops up; you can use Ctrl+J to select the next string of the current string, and Ctrl+K to select the previous string of the current string. Use Tab/Enter or the mouse to select the completion entry as the target. If there is no valid completion content for the input completion, you can continue to input directly, and the corresponding input content will be directly used as input. Any invisible characters are treated as unmatched. When canceling, you need to re-enter the completion content from the original completion process.

5.6 Error modification

By default, the edited content will be grammatically analyzed. If there is a grammatical error, the first error will be displayed in red font, and the subsequent ones will be displayed as an underlined display effect. You can use the **goto error** menu under the **edit** menu to go to the error location for error correction.

The error check in editing is only for syntax check, and the corresponding SQL execution structure may still fail. After executing sql, you can view the corresponding error information through the log window. For more complex error messages, you need to analyze SQL and data content for troubleshooting.

5.7 Save and load

The content of the sql editor will be saved every time the software is normally closed, and the corresponding editing window will be opened automatically when it is reopened. The sql content can also use the file function of **save/open** in the **file** menu. The corresponding file uses UTF-8 encoding and \n line break.

6 Common problems

6.1 Database cannot be connected

This product has a password verification requirement for connecting to the memory database server, please make sure 1. The server works normally and can access the service port of the target server through a normal network connection. 2, Your password is valid and has not been set in the blacklist. 3. The server may exceed the maximum number of connections. See the database server reference description for details.

6.2 Data table display is very slow

Table data will display all the data when the table query is modified, which will cause your data window to display too much content and cause too much slowness. Therefore, it is recommended that you query or modify the display data content to use certain range limiting conditions.

For large amounts of data modification, we recommend that you use standard SQL statements to modify. For a large number of sql statement files, we recommend that you use the **execute sql** menu to execute the script.

6.3 What is the difference between table save (or load), import or (export) and backup (or restore)

Table saving (or loading) is to save (or load from) the entire table data as a binary or sql text file, and the original table cannot exist. Import (or export) is to import (or export) external (or part of the table query) csv/csvs data. The table file must exist and the structure of the import (or export) file must be consistent. Backup (or restore) is an operation for the database, which is to save (or restore) all data tables

6.4 What is the difference between a csv file and a csvs file

csv is a text data file separated by regular commas (or other definitions) and ending with \n. In order to avoid splitting symbols in the text or the end of the line, csvs will display the string as a hexadecimal memory code value (corresponding to the output of the raw2hex function), which can avoid data errors caused by the splitter. When saving, we recommend using the csvs format first. .

6.5 What is immediate mode

Immediate mode is for data modification that often accidentally uses filtering or post-conditions to cause the modified content to be lost. Compared with non-immediate mode, immediate mode always saves every modified data to the database immediately. Therefore, the immediate mode is suitable for modifying the field value of the target table content.

6.6 Code editing is slow

When editing the sql of this product, intelligent analysis and prompting of large sql statements will cause slow code editing. Therefore, we recommend that you do not enable smart prompts when writing large amounts of data, or do not use too long SQL statements.

6.7 Where is the complete sql specification

This manual does not contain the complete sql specification content, you can access the online version of SCT sql specification through the website.

6.8 Why is the sql specification different from the ISO standard SQL

The query language of SCT in-memory database is a query language designed by SCT. It standardizes the query processing flow and removes some syntax that may cause performance problems and increase complexity. The purpose is to retain the ability of data query language and to reduce the cost of learning and use to a greater extent.

6.9 How to update

This product can be automatically updated through the **update** function under the **Help** menu. By default, if our product has a new version and you are within the product service period, you can update the website through information prompts, or contact sales staff for product updates.

6.10 Is this product free

This product is a SCT memory database client, and the product can be used free of charge. However, the distribution and authorization of commercial activities requires you to communicate with us. We ensure that the product has not been modified and adjusted related business services.

6.11 How to report a problem

This product has been fully tested, but there may still be some hard-to-find problems. If you find related problems in use, You can send us the relevant product information via email admin@shucantech.com. We value your feedback and opinions very much.

7 Contact Us

7.1 About Us

Beijing ShuCan Technology Co., Ltd. is a computer technology company located in Beijing, China, focusing on basic technical services. The company was established in April 2019 and consists of a group of teams who love technology research and development. At present, shucan memory database, data database, and shucanWeb server software products have been formed. At the same time, free knowledge retrieval services are provided online. We believe that our products can provide assistance for your information storage, management, and retrieval.

7.2 Contact Information

Online website: http://www.shucantech.com/

Product introduction: http://www.shucantech.com/zh/trial.html

Email: admin@shucantech.com

Welcome product consultation or technical cooperation! Thank you for your attention!