SQL

**SpreadCommander** includes a control to work with **SQL** queries. **SpreadCommander** contains optimized connections for *MS SQL Server*, *MySQL* and *SQLite* providers; it also allows to connect to *ODBC* and *OLEDB* providers.

**SpreadCommander** provides *cmdlets* to simplify work with databases.

Connectuons (connection strings) can be stored with *cmdlet* *Set-DbConnenection* and retrieved by name with *cmdlet* *Get-DbConnection*. Although **SpreadCommander** encrypts connection strings, it is an application with open source code so storing passwords this way is not completely reliable, when possible it is recommended to use *Windows authentication*.

*Cmdlet* *Get-DbConnection* allows to specify not only name of saved connection but also simplified connection string. SQLite allows to specify file path under project directory with '~\' (i.e. '~\Data\mydb.sqlite' is file mydb.sqlite in folder Data under project's root directory. Supported custom connection strings are:

|  |  |
| --- | --- |
| **sqlite:** | *SQLite* database. First part of connection string is *Data Source*. For example 'sqlite:~\Data\MyDb.sqlite' opens connection to database located in file Data\MyDb.sqlite under current project's directory. |
| **mssql:** | Prefix for *Microsoft SQL Server* database. First part of connection string is *Database*. Default server can be stored in settings; if User ID is provided - it will be used, if no - Windows authentication will be used. Connection to default server allows to not specify it in connection string. For example 'mssql:Northwind' opens connection to database Northwind located at default server (it has to be configured in settings) using Windows authentication; 'mssql:Northwind;Server=RemoteServer\ServerName;User ID=user;Password=pwd' connects to database Northwind at server specified in connection string. |
| **mysql:** | Prefix for *MySQL* database. First part of connection string is *Database*. Default server can be stored in settings. Connection to default server allows to not specify it in connection string. For example 'mysql:world' opens connection to database World located at default server (it has to be configured in settings); 'mysql:World;Server=127.0.0.1;User ID=user;Password=pwd' connects to database World at server specified in connection string. |
| **odbc:** | Prefix for *ODBC* database. Requires full connection string. |
| **oldeb:** | Prefix for *OLEDB* database. Requires full connection string. |

Current script is accompanied with script file ReadMe.ps1 with source code.

Extended SQL syntax

**SpreadCommander** supports extended syntax for *SQL queries*. In addition to *SQL* commands that will be sent to the *SQL Server* and executed, **SpreadCommander** supports *client-side commands*.

*Server-side commands* determine which data will be returned from the *SQL Server*.

*Client-side commands* determine how data will be presented at client side. This includes table names, formatting, relations (master-detail) between tables. *Table* here means table in a *DataSet* at client, not table in database.

Look at script *SampleData.sql* in this project for an example for *client-side commands* in SQL script. *Client-side commands* are replaced with spaces before sending command to *SQL Server* and will work even if *SQL Server* does not support comments.

*Client-side commands* start with prefix --# and continue to end of line, or embedded into block /\*#...#\*/ (where '...' is command text). In latter case commands are separated with semicolor.

*Client-side commands* can be located anywhere in *script command* (i.e. anywhere between GO lines). Table name assigned with command *TABLE* will be applied to first, second etc. result set returned from *script command*, not necessarry to result set that gramatically follows command *TABLE*.

Command TABLE

First command is *TABLE*. It allows to specify table in at client side. This table name will be displayed as tab header in *Data grid* and it is used in other commands (*FORMAT CONDITION*, *RELATION*) to refer current table. Command *TABLE* accepts properties *GroupBy* and *OrderBy*, these properties define how data will be shown in **Grid**.

--#table Series with GroupBy="Topic Group", OrderBy="Topic Part desc";

Command FORMAT CONDITION

Command *FORMAT CONDITION* allows to specify conditional formatting for data returned from *SQL Server*.

--#format table Series column [Indicator Name] for "Contains([Indicator Name], 'Agricultural')" with BackColor=Green, ForeColor=White;

--#format table "Energy use - by Regions" column MaxValue with IconSet=Signs3;

To review supported format condition in **Grid** click on button "Formatting" on ribbon tab "Data" when **Grid** is active, or right-click on *column header* and choose *Conditional Formatting/Manage Rules*. Displayed dialog allows to edit existing *format conditions* and add new ones.

Following *format conditions* are supported:

|  |  |
| --- | --- |
| **Expression** | Applies format if row matches specified condition.  --#format table Series column [Indicator Name] for "Contains([Indicator Name], 'Agricultural')" with BackColor=Green, ForeColor=White; |
| **DataBar** | Applies format using a *DataBar*. Bar length changes proprotionally to a cell value.  --#format table [Energy use - by Income] column [AverageValue] with Rule=DataBar, BackColor='LightGreen', BackColor2='LightGray', Gradient=Horizontal |
| **IconSet** | *IconSet* allows to classify values into ranges and display specific icon according to the range.  --#format table "Energy use - by Regions" column MaxValue with IconSet=Signs3; |
| **ColorScale** | Allows to disploay data distribution and variation using a gradation of 2 or 3 colors.  #format table [Energy use - by Income] column [AverageValue] with ColorScale='Red,LightGreen,White' |
| **AboveAverage** **BelowAverage** **AboveOrEqualAverage** **BelowOrEqualAverage** **Unique** **Duplicate** | Applies format if value corresponds specified condition.  --#format table "Energy use" column Value with Rule=AboveAverage, BackColor=LightGreen |
| **Top** **Bottom** | Applies format if value belongs to top or bottom N values. N can be a number or a percent.  --#format table "Energy use" column Value with Rule=Top, Rank='10%', BackColor=LightGreen |
| **DateOccuring** | Applies a format if value refers to specific date or date interval relative to *today*. Day intervalus include Beyond, BeyondThisYear, Earlier, EarlierThisMonth, EarlierThisWeek, EarlierThisYear, Empty, LastWeek, LaterThisMonth, LaterThisWeek, LaterThisYear, MonthAfter1, MonthAfter2, MonthAgo1, MonthAgo2, MonthAgo3, MonthAgo4, MonthAgo5, MonthAgo6, NextWeek, PriorThisYear, SpecificDate, ThisMonth, ThisWeek, Today, Tomorrow, User, Yesterday  --#format table [Table1] column [Value] with DateOccuring=Today, BackColor=Yellow |
| **Rule** **Comparison** | Applies a format if value meets specified comparison condition. Comparison types include Between, Equal, Expression, Greater, GreaterOrEqual, Less, LessOrEqual, NotBetween, NotEqual.  --#format table Regions column [Table Name] with Condition=Equal, Value1='OECD members', BackColor=Green, ApplyToRow=true |

Most *format conditions* (except DataBar, IconSet and ColorScale) support properties that specify *appearance* These properties include:

* BackColor
* BackColor2 (to use with *Gradient*
* BorderColor
* Font
* Gradient (requires *BackColor2*, one of following values: Horizontal, Vertical, ForwardDiagonal, BackwardDiagonal)

Command COMPUTED COLUMN

**SpreadCommander** allows to add client-side computed columns to resulset of SQL queries. To edit computed columns click on button "Computed columns" on ribbon tab "Data" when **Grid** is active.

Syntax is COMPUTED COLUMN [ColumnName] ColumnType IN [TableName] = 'Expression'. ColumnType is one of: STRING (aliases - VARCHAR, CHAR, TEXT), INTEGER (alias INT), DECIMAL (aliases NUMERIC, DOUBLE, FLOAT), BOOLEAN (alias BOOL).

--#computed column [Topic Part] string in Series = "REGEXMATCH([Topic], '(?<=:).\*')"

Command RELATION

**SpreadCommander** executes *SQL queries* into *ADO.Net DataSet* and allows to specify client-side *master-detail relations* between tables.

Syntax for *Relation* is: RELATION [RelationName] [ParentTable] ([ParentColumn(s)]) - [ChildTable] ([ChildColumn(s)]. Multiple column names are comma-separated.

--#relation Rel\_Regions\_EnergyUse [Regions] ([Short Name]) - [Energy use - by Regions] ([Region])

Command CONNECTION

**SpreadCommander** allows to specify different connections for different *SQL queries* (part of script separated by line GO). Connection can be one defined in list of common connections (can be set with cmdlet *Set-DbConnection*) or connection string can be constructed for *Microsoft SQL Server*, *MySQL*, *SQLite*, *ODBC* or *OLEDB* providers. Details see at beginning of this ReadMe file, in section **SQL**. If first query (part of script until first line GO) specifies *Connection* - it will be used by default to execute script, show execution plan and other features.

--#connection "sqlite:~\..\Data\WorldData.db"

Extended SQLite functions

**SpreadCommander** adds extended set of functions to *SQLite*. These functions allow basic data analysis in *SQL queries*.

Scalar functions

Hash

|  |  |
| --- | --- |
| **MD5** | Computed *MD5* hash. |
| **SHA1** | Computes *SHA1* hash. |
| **SHA256** | Computes *SHA256* hash. |

Math

|  |  |
| --- | --- |
| **Acos** | Computes *acos* function. |
| **Asin** | Computes *asin* function. |
| **Atan** | Computes *atan* function. |
| **Ceiling** | Computes *Ceiling* function. |
| **Cos** | Computes *cos* function. |
| **Cosh** | Computes *cosh* function. |
| **Exp** | Computes *exp* function. |
| **Floor** | Computes *floor* function. |
| **IEEERemainder** | Computes *IEEEReminder* function. |
| **Log10** | Computes *log10* function. |
| **Log** | Computes *log* function. |
| **Pow** | Computes *power* function. |
| **Sign** | Computes *sign* function. |
| **Sin** | Computes *sin* function. |
| **Sinh** | Computes *sinh* function. |
| **Sqrt** | Computes *sqrt* function. |
| **Tan** | Computes *tan* function. |
| **Tanh** | Computes *tanh* function. |
| **Truncate** | Computes *truncate* function. |

Path

|  |  |
| --- | --- |
| **PathChangeExtension** | Changes extension of provided filename. |
| **PathCombine** | Combines parts of the path. |
| **PathGetDirectoryName** | Returns directory name of the path. |
| **PathGetExtension** | Returns extension of the path. |
| **PathGetFileName** | Returns file name of the path. |
| **PathGetFileNameWithoutExtension** | Returns file name without extension of the path. |

Random

|  |  |
| --- | --- |
| **RandNormal** | Returns random number with normal distribution. |
| **RandTriangular** | Returns random number with triangular distribution. |
| **RandUniform** | Returns random number with uniform distribution. |

Regular Expressions

|  |  |
| --- | --- |
| **RegexIsMatch** | Returns 1 if string matches provided regular expression pattern and 0 otherwise. Parameters are input string and pattern. |
| **RegexMatch** | Searches the specified input string for the first occurrence of the regular expression. Parameters are input string, pattern and optional match number. |
| **RegexNamedMatch** | Searches the specified input string for the first occurrence of the regular expression and returns value of named group. Parameters are input string, pattern, group name and optional match number. |
| **RegexReplace** | In a specified input string, replaces all strings that match a specified regular expression with a specified replacement string. Parameters are input string, pattern and replacement. |

String

|  |  |
| --- | --- |
| **StringFormat** | Formats string in .Net style. |

Aggreagate functions

|  |  |
| --- | --- |
| **AVG\_GEOM** | Evaluates the geometric mean. |
| **AVG\_HARMONIC** | Evaluates the harmonic mean. |
| **CORR** | Computes the Pearson Product-Moment Correlation coefficient. |
| **CORR\_SPEARMAN** | Spearman correlation. |
| **COVAR** | Estimates the unbiased population covariance from the provided samples. On a dataset of size N will use an N-1 normalizer (Bessel's correction). |
| **COVARP** | valuates the population covariance from the provided full populations. On a dataset of size N will use an N normalizer and would thus be biased if applied to a subset. |
| **STDEV** | Estimates the unbiased population standard deviation from the provided samples. On a dataset of size N will use an N-1 normalizer (Bessel's correction). |
| **STDEVP** | Evaluates the variance from the provided full population. On a dataset of size N will use an N normalizer and would thus be biased if applied to a subset. |
| **VAR** | Estimates the unbiased population variance from the provided samples. On a dataset of size N will use an N-1 normalizer (Bessel's correction). |
| **VARP** | Evaluates the variance from the provided full population. On a dataset of size N will use an N normalizer and would thus be biased if applied to a subset. |
| **InterQuantileRange** | Estimates the inter-quartile range from the provided samples. Approximately median-unbiased regardless of the sample distribution. |
| **Kurtosis** | Estimates the unbiased population kurtosis from the provided samples. Uses a normalizer (Bessel's correction; type 2). |
| **KurtosisP** | Evaluates the kurtosis from the full population. Does not use a normalizer and would thus be biased if applied to a subset (type 1). |
| **Median** | Estimates the sample median from the provided samples. |
| **Skewness** | Estimates the unbiased population skewness from the provided samples. Uses a normalizer (Bessel's correction; type 2). |
| **SkewnessP** | Evaluates the skewness from the full population. Does not use a normalizer and would thus be biased if applied to a subset (type 1). |
| **Quantile** | Estimates the tau-th quantile from the provided samples. The tau-th quantile is the data value where the cumulative distribution function crosses tau. Approximately median-unbiased regardless of the sample distribution. |
| **RMS** | Evaluates the root mean square (RMS) also known as quadratic mean. |

Collations

|  |  |
| --- | --- |
| **Logical** | Logical collation. Strings '1', '2' .. '10', '20' are ordered according numeric values, not 1, 10, 2, 20. Case sensitive. |
| **LogicalCI** | Logical collation. Strings '1', '2' .. '10', '20' are ordered according numeric values, not 1, 10, 2, 20. Case insensitive. |

SQL cmdlets

This section contains help for *cmdlets* that allow to output data (not only results of executing *SQL queries*) into **Data** tab.

Cmdlet *Out-Data*

**NAME**  
    Out-Data  
      
**SYNTAX**  
    Out-Data [[-DataRecord] <psobject>] [-DataSource <Object>] [-TableName <string>] [-IgnoreErrors] [-Formatting <string>] [-Replace] [-RelationName <string>] [-ParentTable <string>] [-ParentColumns <string[]>] [-ChildColumns <string[]>] [-PassThru] [-ResetDataSet] [<CommonParameters>]  
      
      
**PARAMETERS**  
**-ChildColumns <string[]>**  
        Create master-detail relationship. List of column names for relation. If not set - ParentColumns are using.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-DataRecord <psobject>**  
        Data source for data tables. Data source shall implement interface IList or IListSource and final IList shall implement ITypedList.  
          
        Required?                    false  
        Position?                    0  
        Accept pipeline input?       true (ByValue)  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-DataSource <Object>**  
        Data source  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-Formatting <string>**  
        Format conditions for spreadsheet table  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-IgnoreErrors**  
        Ignore errors thrown when getting property values  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      NoErrors  
        Dynamic?                     false  
          
**-ParentColumns <string[]>**  
        Create master-detail relationship. Comma-separated list of parent columns  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-ParentTable <string>**  
        Master table for master-detail relationship  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-PassThru**  
        Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-RelationName <string>**  
        Name of master-detail relation  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-Replace**  
        Replace existing table if it exists.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-ResetDataSet**  
        When set - current tables, relations etc. will be removed from the DataSet.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      reset  
        Dynamic?                     false  
          
**-TableName <string>**  
        Table name  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      Table  
        Dynamic?                     false  
          
**<CommonParameters>**  
        This cmdlet supports the common parameters: Verbose, Debug,  
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,  
        OutBuffer, PipelineVariable, and OutVariable. For more information, see  
        about\_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).   
      
      
**INPUTS**  
    System.Management.Automation.PSObject  
      
      
**OUTPUTS**  
    System.Object  
      
**ALIASES**  
    None  
      
  
**REMARKS**  
    None

Cmdlet *Out-DataSet*

**NAME**  
    Out-DataSet  
      
**SYNTAX**  
    Out-DataSet [[-DataSource] <DataSet>] [-Replace] [-Formatting <string>] [-ResetDataSet] [-NoCopyTables] [-PassThru] [<CommonParameters>]  
      
      
**PARAMETERS**  
**-DataSource <DataSet>**  
        ADO.Net DataSet for Data tab.  
          
        Required?                    false  
        Position?                    0  
        Accept pipeline input?       true (ByValue)  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-Formatting <string>**  
        Format conditions for spreadsheet table  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-NoCopyTables**  
        When set - tables are removed from DataSource. This is more efficient but DataSource cannot be used later.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-PassThru**  
        Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-Replace**  
        Replace existing tables if they exist.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      None  
        Dynamic?                     false  
          
**-ResetDataSet**  
        When set - current tables, relations etc. will be removed from the DataSet.  
          
        Required?                    false  
        Position?                    Named  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      reset  
        Dynamic?                     false  
          
**<CommonParameters>**  
        This cmdlet supports the common parameters: Verbose, Debug,  
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,  
        OutBuffer, PipelineVariable, and OutVariable. For more information, see  
        about\_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).   
      
      
**INPUTS**  
    System.Data.DataSet  
      
      
**OUTPUTS**  
    System.Object  
      
**ALIASES**  
    None  
      
  
**REMARKS**  
    None

Cmdlet *Remove-DataTable*

**NAME**  
    Remove-DataTable  
      
**SYNTAX**  
    Remove-DataTable [-TableName] <string> [<CommonParameters>]  
      
      
**PARAMETERS**  
**-TableName <string>**  
        Table name  
          
        Required?                    true  
        Position?                    0  
        Accept pipeline input?       false  
        Parameter set name           (All)  
        Aliases                      Table  
        Dynamic?                     false  
          
**<CommonParameters>**  
        This cmdlet supports the common parameters: Verbose, Debug,  
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,  
        OutBuffer, PipelineVariable, and OutVariable. For more information, see  
        about\_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).   
      
      
**INPUTS**  
    None  
      
      
**OUTPUTS**  
    System.Object  
      
**ALIASES**  
    None  
      
  
**REMARKS**  
    None

Cmdlet *Clear-Data*

**NAME**  
    Clear-Data  
      
**SYNTAX**  
    Clear-Data [<CommonParameters>]  
      
      
**PARAMETERS**  
**<CommonParameters>**  
        This cmdlet supports the common parameters: Verbose, Debug,  
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,  
        OutBuffer, PipelineVariable, and OutVariable. For more information, see  
        about\_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).   
      
      
**INPUTS**  
    None  
      
      
**OUTPUTS**  
    System.Object  
      
**ALIASES**  
    None  
      
  
**REMARKS**  
    None

Table of Contents

SQL 1

Extended SQL syntax 3

Command TABLE 3

Command FORMAT CONDITION 3

Command COMPUTED COLUMN 5

Command RELATION 5

Command CONNECTION 5

Extended SQLite functions 7

Scalar functions 7

Hash 7

Math 7

Path 8

Random 8

Regular Expressions 8

String 9

Aggreagate functions 9

Collations 10

SQL cmdlets 11

Cmdlet Out-Data 11

Cmdlet Out-DataSet 14

Cmdlet Remove-DataTable 16

Cmdlet Clear-Data 17

Table of Contents 18