Release notes for version 3.5

Important! Create a backup copy of your projects before updating to the new version. Projects saved in the new version can't be opened in earlier versions.

Behavior changes

1. **Remove()** function now returns the original value if nothing is removed.

Old behavior	New behavior
Remove() converts non-text values to text even if nothing is removed. Example: remove(123, 4, 5) returns '123' (text).	Remove() does NOT convert non-text values to text if nothing is removed. Example: remove(123, 4, 5) returns 123 (number).

2. **Keepbefore()** function now returns the original value if the substring wasn't found.

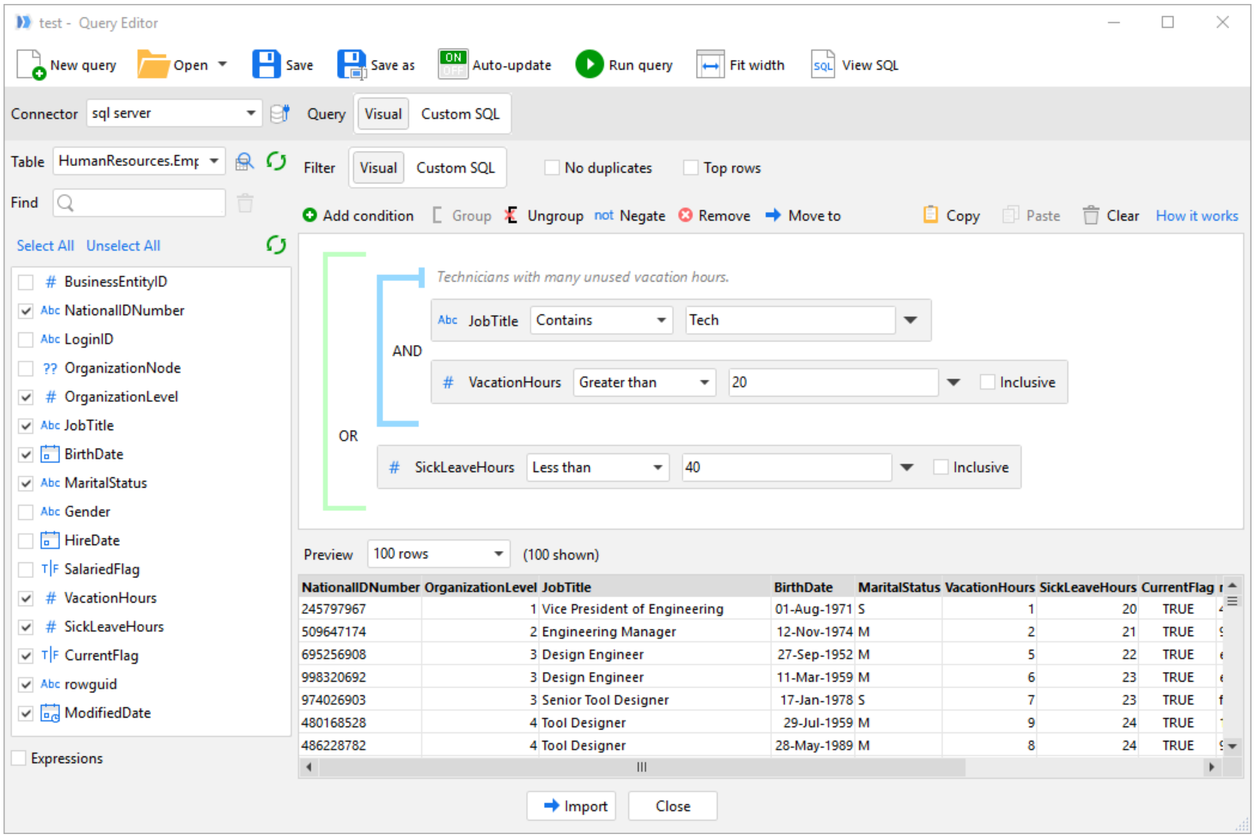
Old behavior	New behavior
Keepbefore() returns empty value if no match happened. Example: keepbefore('123',4') returns empty value.	Keepbefore() returns the original value if no match happened. Example: keepbefore('123',4') returns '123'.

What's new

Visual condition builder

The previous version introduced *queries* and Query Editor. With this release queries get significant improvement which is the Visual Condition Builder. It allows composing filtering conditions visually, creating complex conditions using AND and OR groups. Conditions can be created either by dragging field names from the field selector on the left or from the data grid below. You can also double-click a field header in the data grid and see unique values in the field, or select values to filter. Other features include:

- Either the visual condition or the entire query can be converted into custom SQL by switching respective Visual/Custom selector.
- Visual conditions of one query can be copied and pasted into another query.
- Annotations can be added to both groups and individual conditions.
- Query result preview is automatically updated on condition change (when Auto-update is on).



Connector: sql server

Table: HumanResources.Emp

☐ BusinessEntityID
☑ NationalIDNumber
☐ LoginID
☐ OrganizationNode
☑ OrganizationLevel
☑ JobTitle
☑ BirthDate
☑ MaritalStatus
☐ Gender
☐ HireDate
☐ SalariedFlag
☑ VacationHours
☑ SickLeaveHours
☑ CurrentFlag
☑ rowguid
☑ ModifiedDate

☐ No duplicates
☐ Top rows

OR
- AND — *Technicians with many unused vacation hours.*
  - JobTitle Contains Tech
  - VacationHours Greater than 20 ☐ Inclusive
- SickLeaveHours Less than 40 ☐ Inclusive

Preview: 100 rows (100 shown)

NationalIDNumber	OrganizationLevel	JobTitle	BirthDate	MaritalStatus	VacationHours	SickLeaveHours	CurrentFlag
245797967	1	Vice President of Engineering	01-Aug-1971	S	1	20	TRUE
509647174	2	Engineering Manager	12-Nov-1974	M	2	21	TRUE
695256908	3	Design Engineer	27-Sep-1952	M	5	22	TRUE
998320692	3	Design Engineer	11-Mar-1959	M	6	23	TRUE
974026903	3	Senior Tool Designer	17-Jan-1978	S	7	23	TRUE
480168528	4	Tool Designer	29-Jul-1959	M	9	24	TRUE
486228782	4	Tool Designer	28-May-1989	M	8	24	TRUE

Screenshot 1: Visual condition builder in Query Editor.

New transformations

File splitter transformation aims to help processing very large text files. It splits a text file into smaller chunks (also text files) that can further be processed using iterations. The chunks can be either of fixed length (by row count), or split by unique values of a column (e.g. one chunk per unique Date). File Splitter is not available in the free edition.

Filter by search transformation keeps or removes rows where values in specified column contain particular substring. It also supports filtering using a regular expression.

Delete database rows transformation deletes rows defined by a query, in a database table. Only the rows that satisfy the filtering condition of the query are deleted. This transformation coupled with **Export to Database** transformation makes it possible to replace particular rows in database tables (i.e. perform updates).

Delete matching database rows transformation deletes rows in specified database table where values in selected key fields match values in selected columns of an EasyMorph table. This transformation can be used, for instance, to remove transactions of customers with particular IDs. The list of IDs can be obtained from another data source (e.g. a text file).

Select matching database rows transformation selects rows in specified database table where values in selected key fields match values in selected columns of an EasyMorph table. This transformation can be used, for instance, to retrieve transactions for a list of customers where the list of customer IDs is obtained from another data source (e.g. a spreadsheet). It can also be helpful for dealing with large tables, where only a few columns (including primary/foreign keys) are fetched first, then filtering is performed using EasyMorph transformations, and then full rows are fetched only for filtered primary/foreign keys thus reducing total imported data volume.

Changes to existing transformations

Match transformation now has Fuzzy Matching mode where values can be matched even if they are not identical (e.g. because of typos). In this mode matching is done by calculating the edit distance (as per Damerau-Levenshtein) – values with an edit distance smaller than specified are matched.

Merge transformation now has Full Join mode which behaves similarly to a full outer SQL join.

Export to Excel transformation now allows creating new sheets in existing spreadsheets. It also attempts to honor column widths when exporting data.

New functions

proper(text) – converts text to the so called “proper case”. Note that abbreviations always remain upper-case. Example:

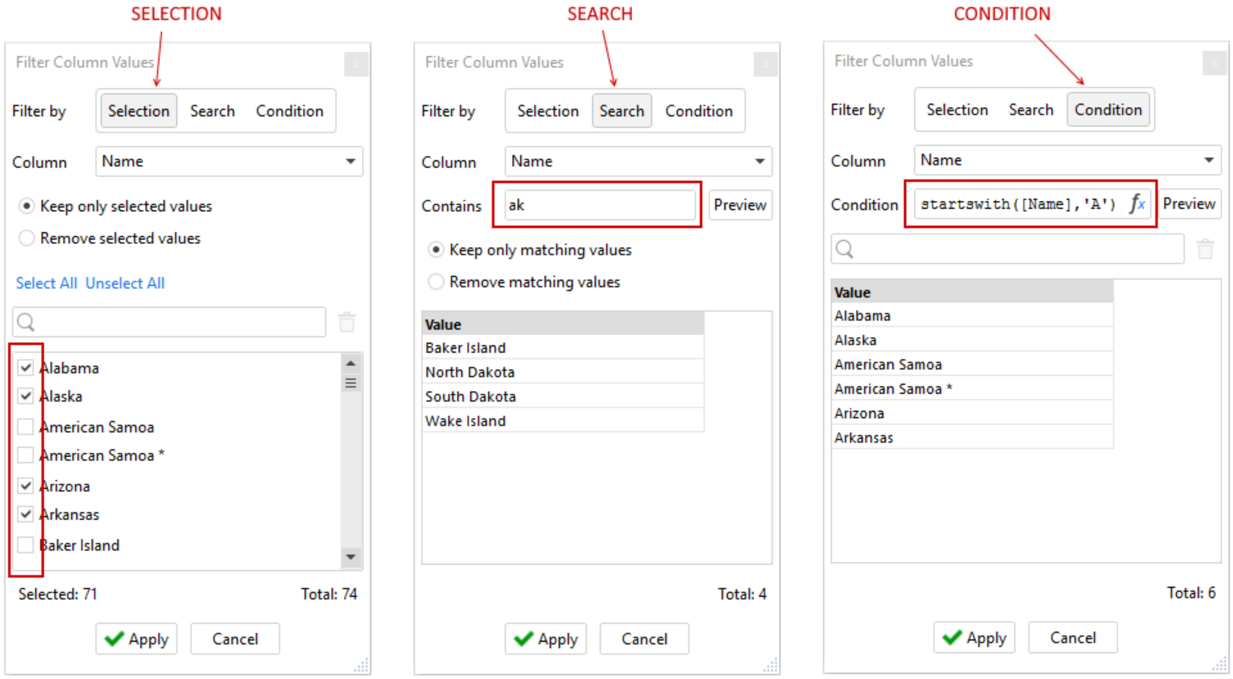
`proper('BAck to tHe USSR’)` returns ‘Back To The USSR’.

sanitize(text) – replaces tabs and line breaks with spaces and removes hidden system characters with ASCII codes 0 to 31.

agemonths(number_date, number_date) – calculates the number of full months between two dates. Example:

`agemonths(makedate(2015,12,01), makedate(2017,1,31))` returns 13.

New Filter Pop-up



Column: Name
● Keep only selected values
○ Remove selected values

☑ Alabama
☑ Alaska
☐ American Samoa
☐ American Samoa *
☑ Arizona
☑ Arkansas
☐ Baker Island

Selected: 71 Total: 74

Column: Name
Contains: ak
● Keep only matching values
○ Remove matching values

Value
Baker Island
North Dakota
South Dakota
Wake Island

Total: 4

Column: Name
Condition: `startswith([Name],'A')`

Value
Alabama
Alaska
American Samoa
American Samoa *
Arizona
Arkansas

Total: 6

Screenshot 2: Filter pop-up (invoked by double-clicking a column header).

Starting from this release double-clicking a column header brings up the new Filter pop-up which allows viewing/searching unique values in the column as well as filtering (according transformation is created automatically). The pop-up has three modes:

- Selection – explicitly select items that should be kept (removed).
- Search – only items that contain particular substring are kept (removed).
- Condition – use expression to filter items. Only items where the expression returns TRUE are kept.

The Unique Values pop-up used in previous versions is discontinued.

Miscellaneous

- Added support for DB2 (LUW) SQL dialect in ODBC connections.
- Added support for Vertica SQL dialect in ODBC connections.
- Added Query Database button to the Main ribbon toolbar.
- Added instant cancellation to 18 more transformations.
- Cell Metadata window now shows formatted number values as well.
- The installer is now signed with a code signing certificate to ensure that it wasn’t tampered with when transferred over the internet.