Нечёткий поиск

Под нечётким поиском понимается поиск по ключевым словам с учетом возможных ошибок в написании слова. Нечёткий поиск применяется при поиске слов с опечатками, а также в тех случаях, когда возникают сомнения в правильном написании - улицы, названия организации, фамилии и т.п.

В Deductor 5.3 (начиная со сборки 5.3.0.71) нечёткий поиск реализуется двумя обработчиками: «***Индекс нечёткого поиска»*** и «***Слияние с индексом нечёткого поиска»,*** а также функциями обработчика ***«Калькулятор»***.

Алгоритмы нечёткого поиска характеризуются *метрикой* — функцией расстояния между двумя словами, позволяющей оценить степень их сходства в данном контексте.

Одними из наиболее известных метрик являются расстояния *Левенштейна* и *Дамерау-Левенштейна*.

Расстояние Левенштейна (также **редакционное расстояние** или **дистанция редактирования**) между двумя строками в теории информации и компьютерной лингвистике — это минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

Пусть S_1 и S_2 — две строки (длиной M и N соответственно) над некоторым алфавитом, тогда редакционное расстояние (расстояние Левенштейна) $\mathrm{d}(S_1, S_2)$ можно подсчитать по следующей рекуррентной формуле:

$\mathrm{d}(S_1, S_2) = D(M, N)$, где

$$D(i,j) = \begin{cases} 0 & ; \ i = 0, \ j = 0 \\ i & ; \ j = 0, \ i > 0 \\ j & ; \ i = 0, \ j > 0 \\ \min(& \\ \quad D(i, j-1) + 1, & \\ \quad D(i-1, j) + 1, & ; \ j > 0, \ i > 0 \\ \quad D(i-1, j-1) + \mathrm{m}(S_1[i], S_2[j]) & \\ ) & \end{cases},$$

где $\mathrm{m}(a,b)$ равна нулю, если $a = b$ и единице в противном случае; $\min(a,b,c)$ возвращает наименьший из аргументов.

Расстояние Дамерау — Левенштейна — это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и транспозиции (перестановки двух соседних символов), необходимых для перевода одной строки в другую. Является модификацией расстояния Левенштейна.

Расстояние Дамерау-Левенштейна между двумя строками a и b определяется функцией $d_{a,b}(|a|,|b|)$как:

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

где $1_{(a_i \neq b_j)}$ - это индикаторная функция равная нулю при $a_i = b_j$ и 1 в противном случае.

Каждый рекурсивный вызов соответствует одному из случаев:

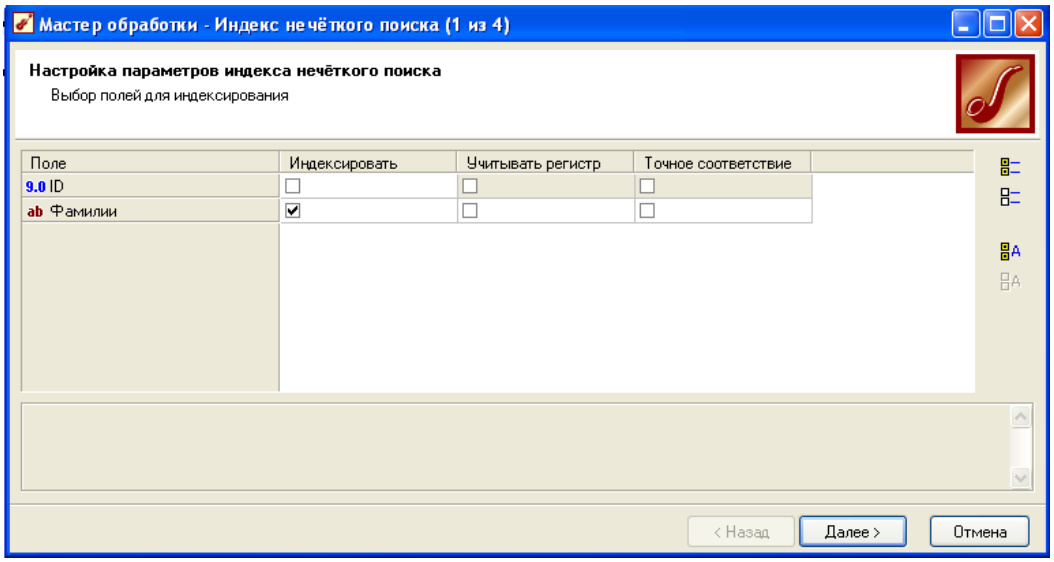
- $d_{a,b}(i-1,j)+1$ соответствует удалению символа (из a в b).
- $d_{a,b}(i,j-1)+1$ соответствует вставке (из a в b).
- $d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)}$ соответствие или несоответствие, в зависимости от совпадения символов
- $d_{a,b}(i-2,j-2)+1$ в случае перестановки двух последовательных символов.

Обработчик «Индекс нечёткого поиска»

Данный обработчик предназначен для построения индекса нечёткого поиска - набора слов, по которым производится поиск. Индекс нечёткого поиска применяется для последующего совместного использования с обработчиком «***Слияние с индексом нечёткого поиска»***.

Настройка обработчика

На первом шаге ***Мастер обработки – Индекс нечёткого поиска*** необходимо определить поля для индексации. Если существует необходимость учитывать регистр, нужно отметить соответствующий пункт.



Также можно выбрать поле для точного поиска. В этом случае при слиянии с комплексным индексом нечёткого поиска необходимо установить соответствия всем проиндексированным полям для точного поиска и минимум одним полем для нечёткого поиска.

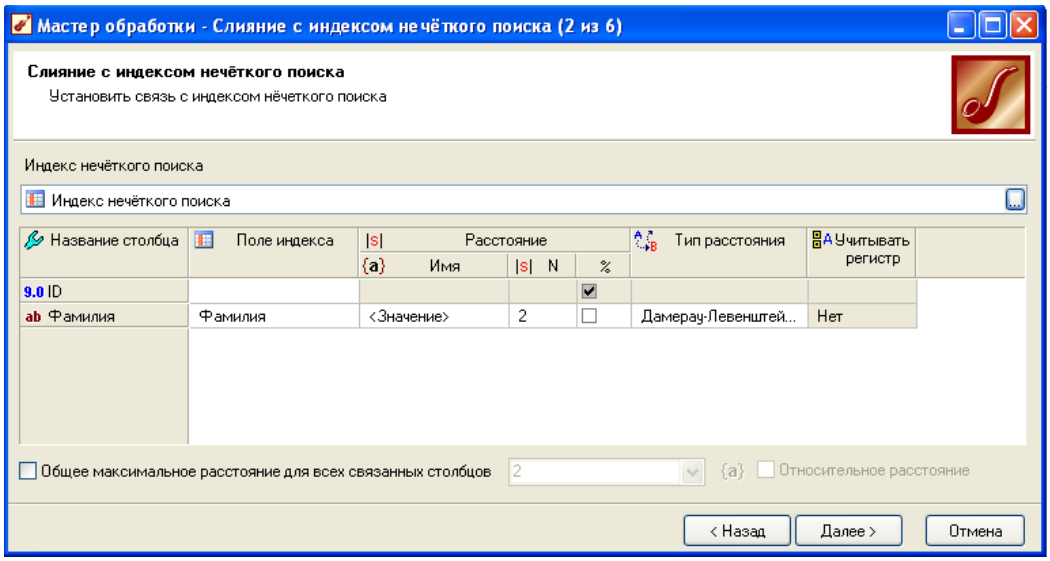
Обработчик «Слияние с индексом нечёткого поиска»

Применяется для выполнения нечёткого поиска. Слияние можно осуществить только с узлом «***Индекс нечёткого поиска***». Тип слияния – внешнее левое соединение.

Настройка обработчика

На первом шаге ***Мастер обработки – Слияние с индексом нечёткого поиска*** необходимо определить узел, с которым осуществляется слияние. Далее устанавливаются параметры нечёткого поиска:

- соответствие столбца полю индекса;
- расстояние между словами, позволяющее оценить степень их сходства. Расстояние можно задать в виде числа, в процентах, в виде переменной. Расстояние в процентах рассчитывается как отношение расстояния между словами к длине исходного слова. Также есть возможность указать общее максимальное расстояние для всех связанных столбцов (или в виде числа, или в процентах, или переменной);
- тип расстояния. Возможен выбор из двух типов – расстояние Левенштейна и расстояние Дамерау-Левенштейна.



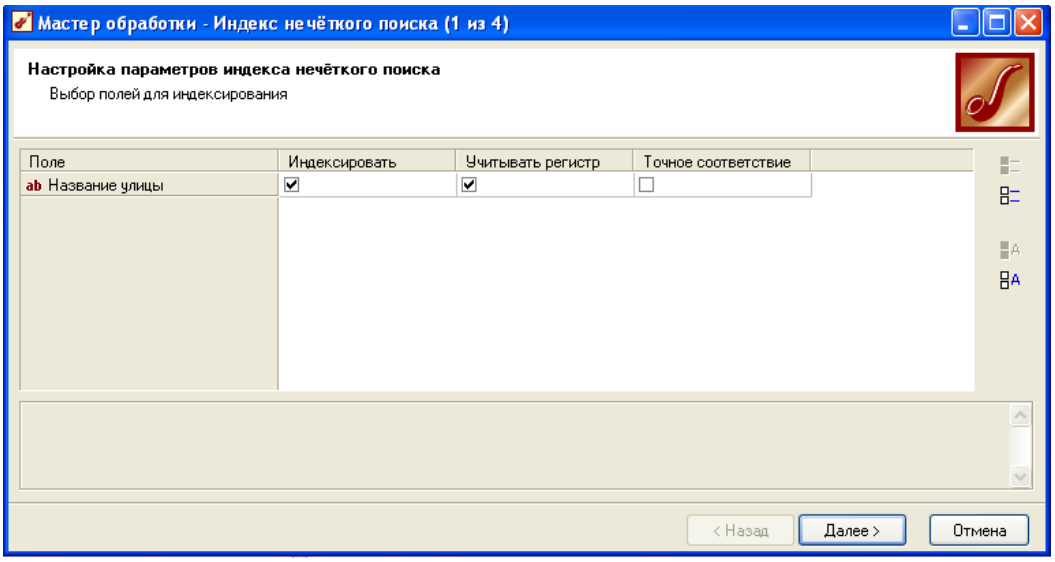
На третьем шаге необходимо указать поля, которые будут включены в выходной набор данных. Для этого щелчком левой кнопки мыши нужно установить

галочку напротив метки поля, которое необходимо включить в выходной набор данных. На этой же странице мастера можно задать имена полей в результирующей таблице и указать, включать ли в выходной набор поля входного набора данных, появляющиеся впоследствии, а также включать ли данные расстояний редактирования.

Пример применения обработчиков нечёткого поиска

Имеем список, состоящий из нескольких названий улиц города, которые могут содержать ошибки. Необходимо оценить правильность написания названий.

Импортируем файл, представляющий собой список улиц города. Данные этого списка рассматриваются как эталонные. Для построения индекса нечёткого поиска используем обработчик ***«Индекс нечёткого поиска»***. При поиске ошибок будем учитывать регистр.

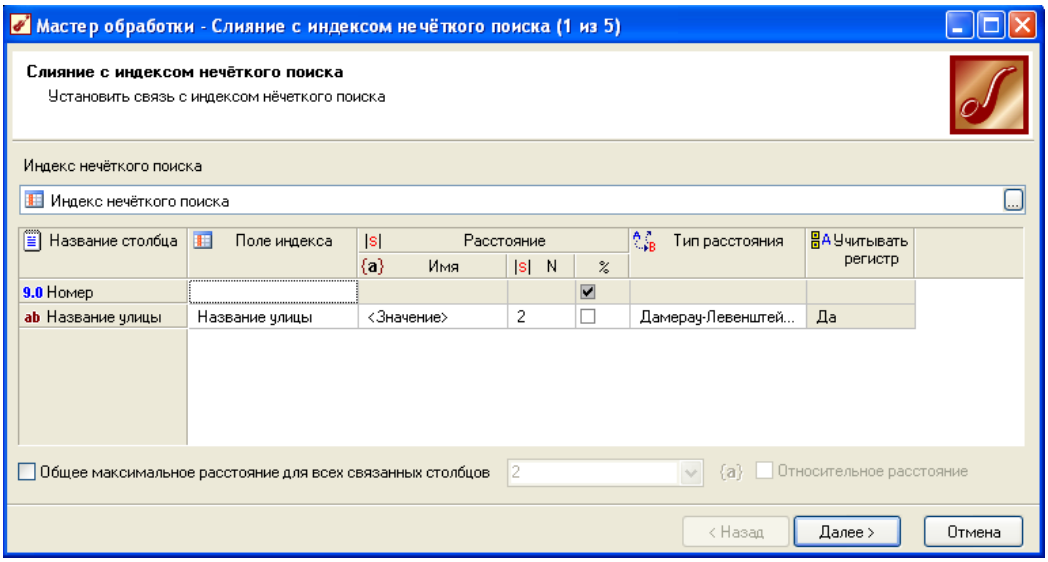


Далее импортируем текстовый файл, данные которого нужно проверить на правильность.

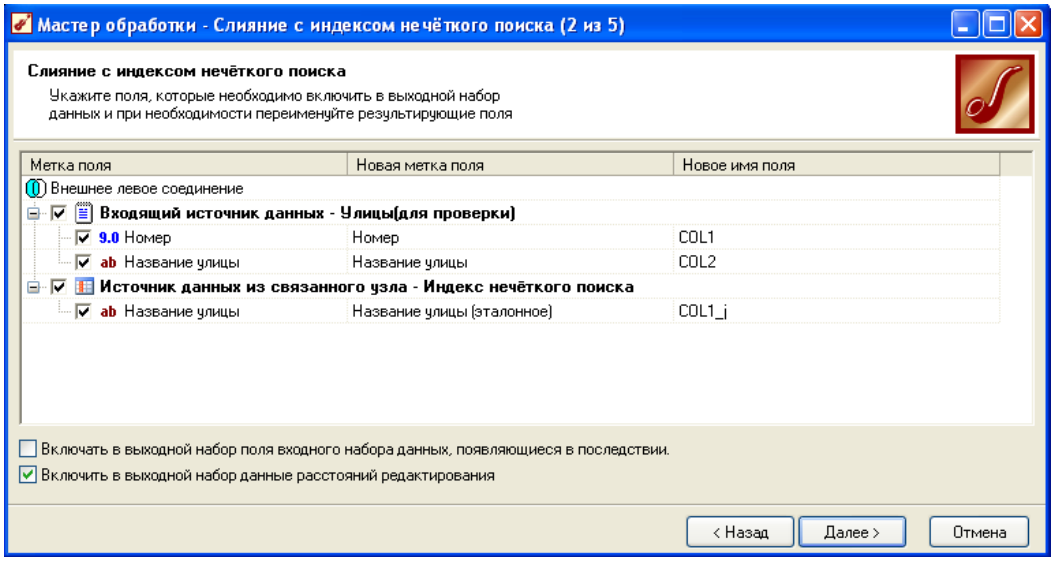
Номер	Название улицы
1	Алейная
2	Веденская
3	держинского
4	Большая
5	Урицкоого
6	Гаранжая
7	Палетаева
8	Гогоголя
9	Северная
10	Семакша

Для обнаружения ошибок и установления соответствия с правильным названием применим обработчик ***«Слияние с индексом нечёткого поиска».***

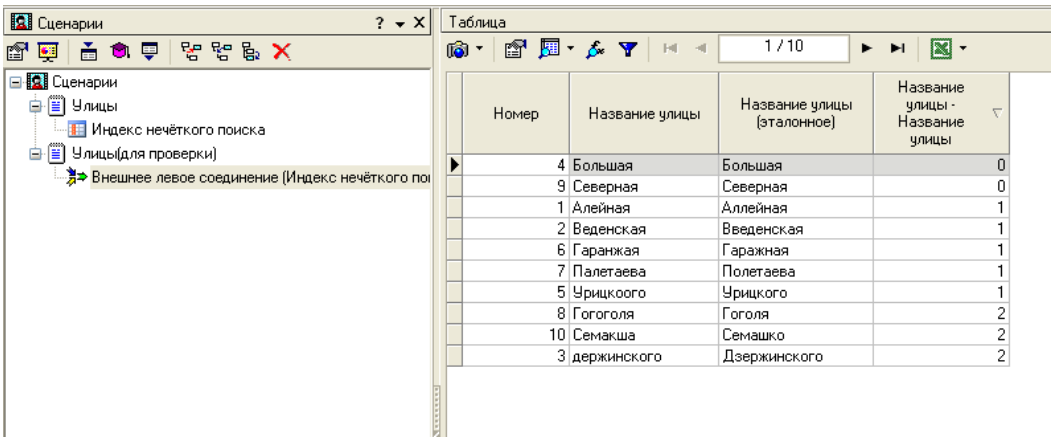
Выбираем узел, с которым осуществляется слияние. Далее установим соответствие столбца полю индекса, максимальное расстояние между столбцами и тип расстояния.



На следующем шаге укажем поля, которые будут включены в выходной набор данных.



В результате получим:



Третий столбец показывает расстояние между исходными и эталонными значениями. Таким образом, мы видим, что первые два названия являются правильными (расстояние равно 0), во всех остальных имеются ошибки.

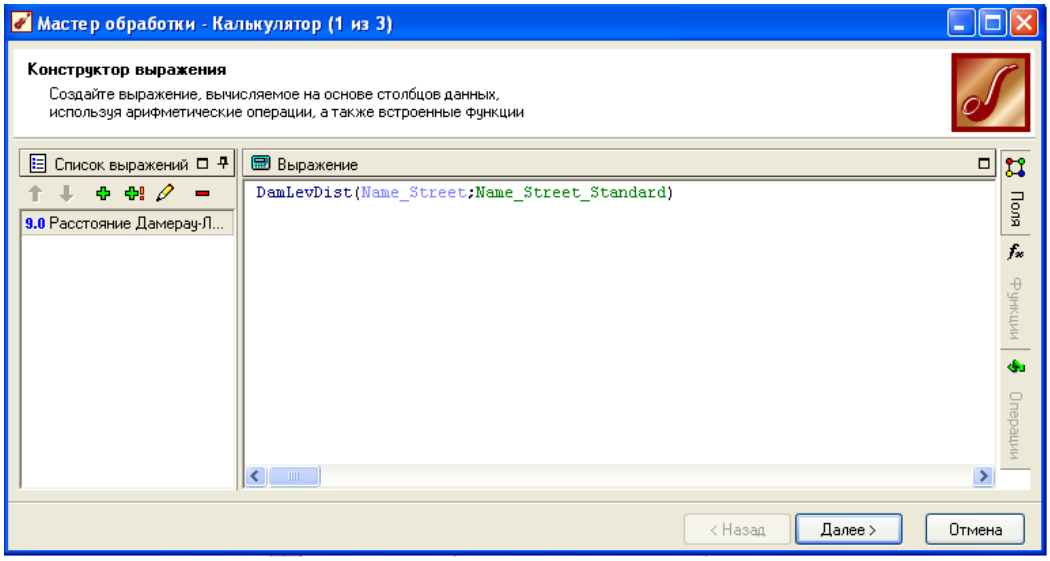
Нечёткий поиск в обработчике «Калькулятор»

Нечёткий поиск также можно реализовать при помощи обработчика ***«Калькулятор»***, используя функции *DamLevDist* (определяет расстояние Дамерау-Левенштейна) и *LevDist* (определяет расстояние Левенштейна).

Рассмотрим реализацию нечёткого поиска с использованием данного обработчика на примере, описанном выше.

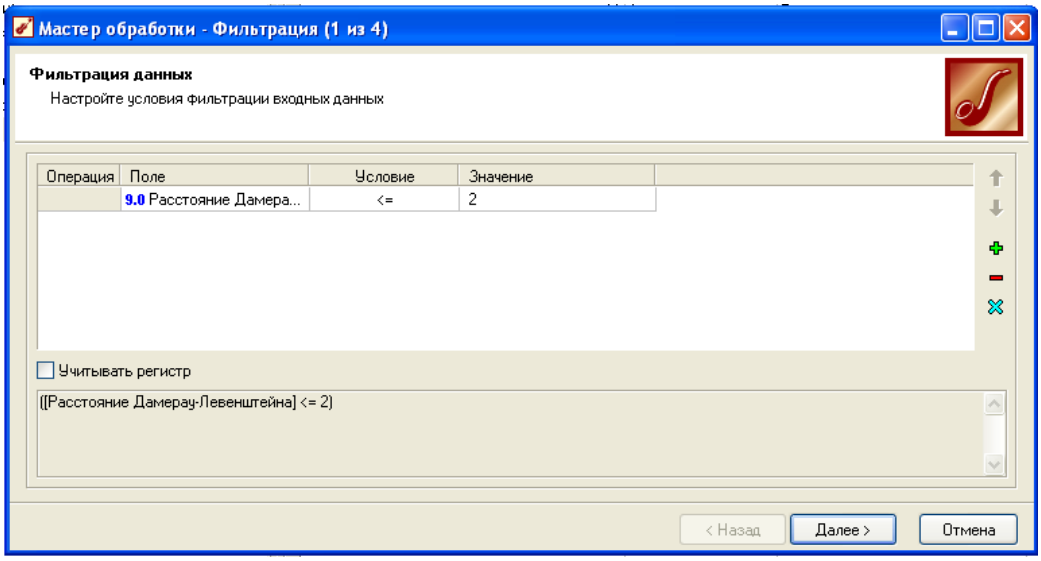
При помощи обработчика ***«Слияние с узлом»*** выполним полное внешнее соединение списка улиц для проверки и эталонного списка названий. В результате получим соединение всех названий улиц из первого списка со всеми названиями из второго списка.

Для вычисления расстояния Дамерау-Левенштейна используем обработчик ***«Калькулятор»***:



В результате получим расстояние Дамерау-Левенштейна, вычисленное по всем возможным парам названий между эталонным и проверяемым наборами.

Далее применим обработчик ***«Фильтрация»***. Фильтрацию осуществляем по столбцу «Расстояние Дамерау-Левенштейна» при условии, что расстояние между названиями не больше 2:



В результате получим таблицу, аналогичную таблице результата, полученную в примере выше:

Название улицы	Название улицы (эталон)	Расстояние Дамерау-Левенштейна
Большая	Большая	0
Северная	Северная	0
Алейная	Аллейная	1
Веденская	Введенская	1
Урицкоого	Урицкого	1
Гаранжая	Гаражная	1
Палетаева	Полетаева	1
держинского	Дзержинского	2
Гогоголя	Гоголя	2
Семакша	Семашко	2

Таким образом, реализации нечёткого поиска двумя способами дали аналогичный результат. При этом второй из способов (использование функций обработчика ***«Калькулятор»***) в данном примере не только содержит больше узлов в сценарии, но и существенно увеличивает необходимый для своей работы набор данных в памяти. Функции нечёткого поиска в *Калькуляторе* рекомендуется применять для реализации сложных правил сравнения или для расчёта вторичных составляющих набора, в которых расстояния Левенштейна/Дамерау-Левенштейна используются как один из компонентов формулы.