

Мастер-класс по подсистеме Инструменты разработчика 2.82.

Документ подготовлен в 2013г Дорохиной Викторией по видеозаписи мастер-класса Старых С.А. с конференции Infostart Event Evolution 2013 и отредактирован Старых С.А. в 2014г.

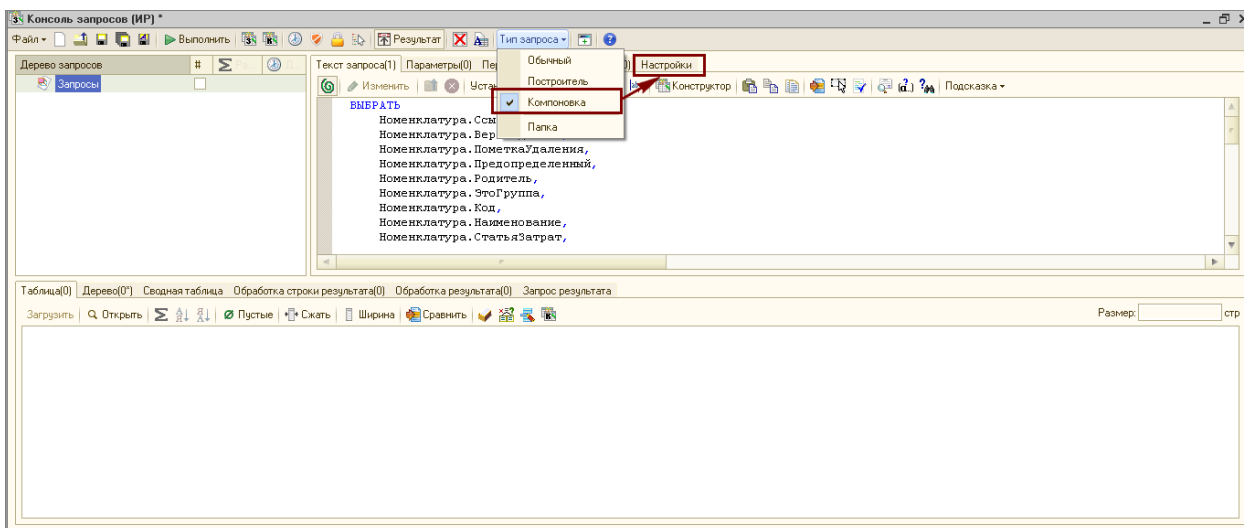
Консоль запросов

Начнем с Консоли запросов. Посмотрим, как в ней можно выбрать и обработать данные.

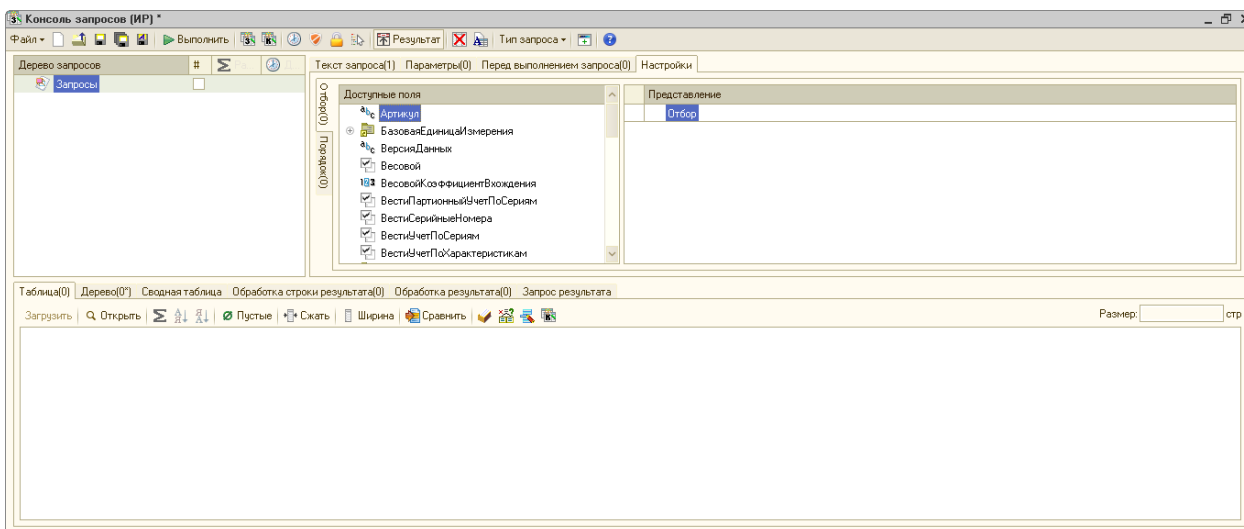
Работа с консолью запросов в режиме «Компоновка»

Пусть у нас есть задача выбрать все услуги из справочника Номенклатура (все элементы со значением реквизита «ВидНоменклатуры» равным «Услуга»). Но, допустим, нам просто лень сначала задавать в тексте запроса отбор, а потом отдельно назначать значения параметров для этого элемента отбора.

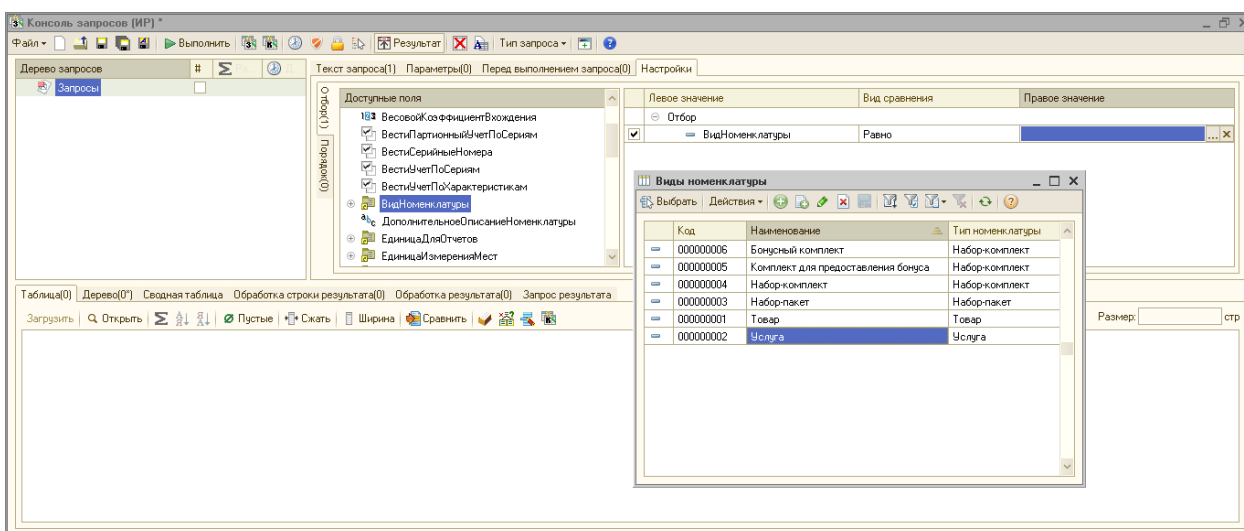
Чтобы упростить свою задачу, мы можем включить тип запроса «Компоновка». Обратите внимание, что в дереве запросов у нас иконка сменилась. И также стала активной закладка «Настройки» – это динамические настройки запроса в режиме построителя либо компоновки.



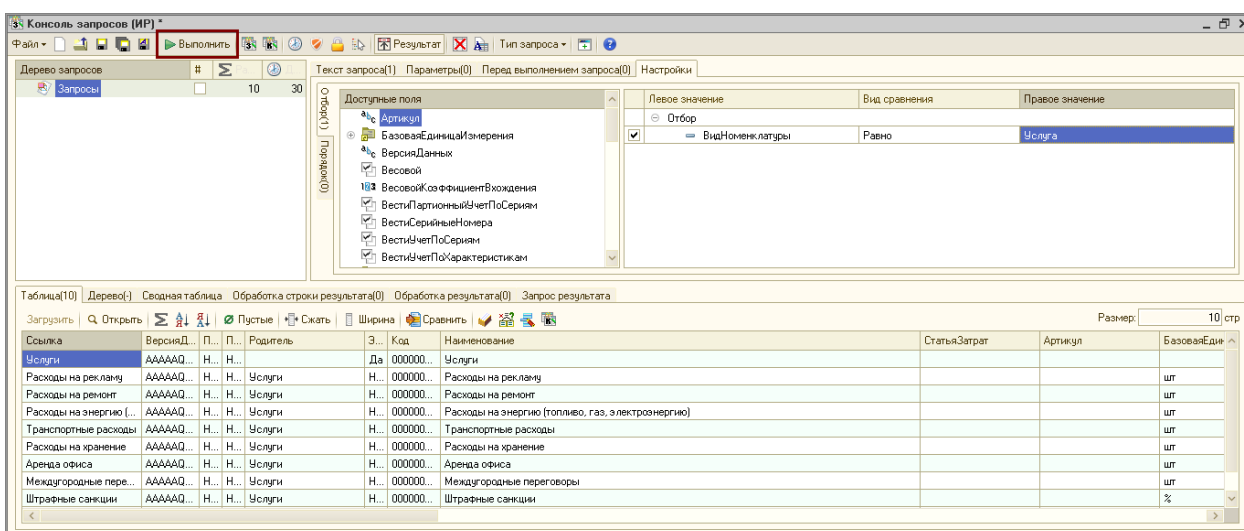
Итак, на закладке «Настройки» мы видим, что у нас появились доступные поля и для них стал доступен обычный отбор компоновки.



Здесь двойным щелчком или перетаскиванием я добавляю в поля отбора поле «ВидНоменклатуры» и выбираю для него значение «Услуга».



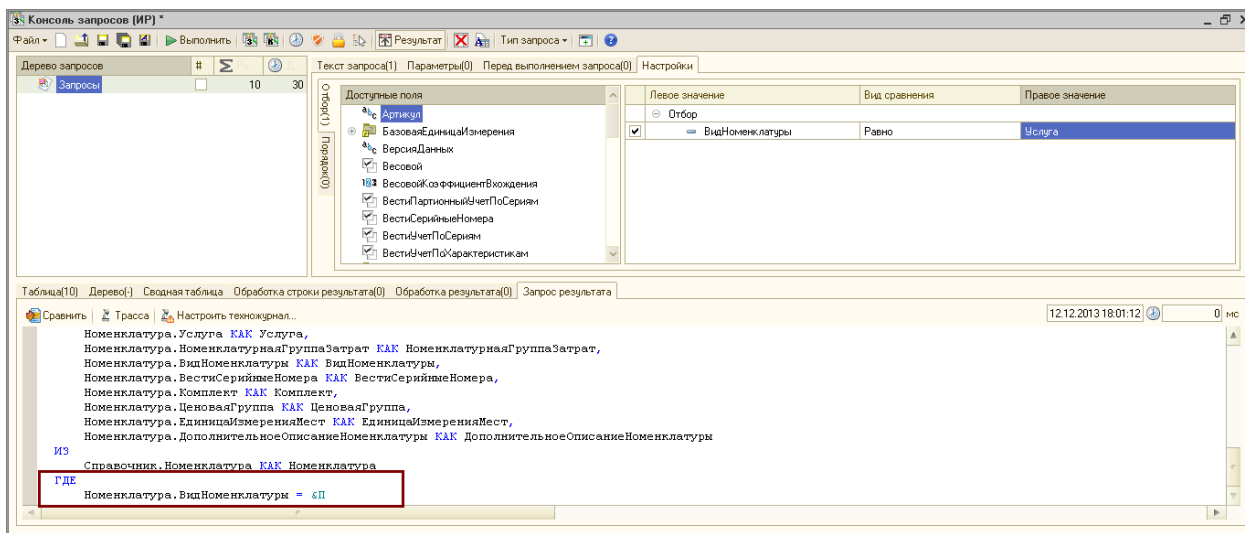
Выполняю запрос, и у меня выбираются только номенклатурные позиции – услуги.



Еще раз обращаю внимание, что в самом запросе мы, вроде как, ничего не настраивали (на закладке «Текст запроса» у нас текст запроса остался тот же самый), а результат выводится уже с фильтрацией.

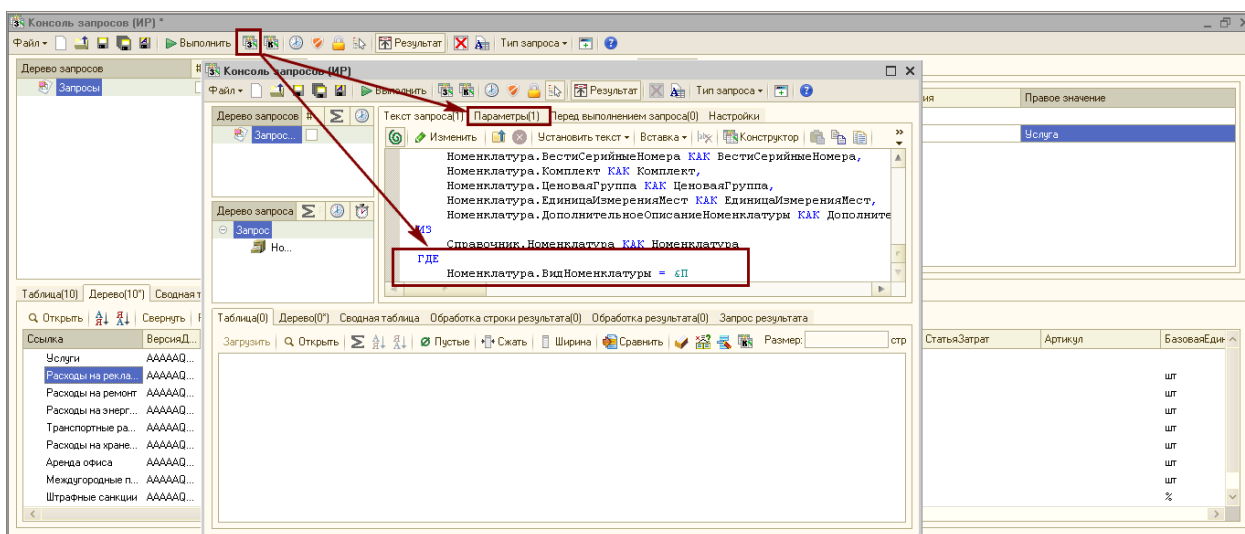
Кстати, текст выполненного результата запроса с учетом всех этих динамических настроек, которые в процессе выполнения дополнительно накладываются – можно смотреть на закладке «Запрос результата».

И если у вас есть какие-то сомнения в том, что отбор наложился ожидаемым вами образом, здесь можно посмотреть, в какое выражение в тексте запроса его преобразовала платформа.



Передача результирующего запроса компоновки в другую консоль запросов

А если вы захотите этот конечный запрос с учетом этих отборов посмотреть в другой консоли запросов, то вы нажимаете кнопку «Открыть исполнение запроса в консоли запросов», где открывается еще один экземпляр консоли запросов, в котором вы уже видите вот этот конечный запрос.



Обработка результата в консоли запросов

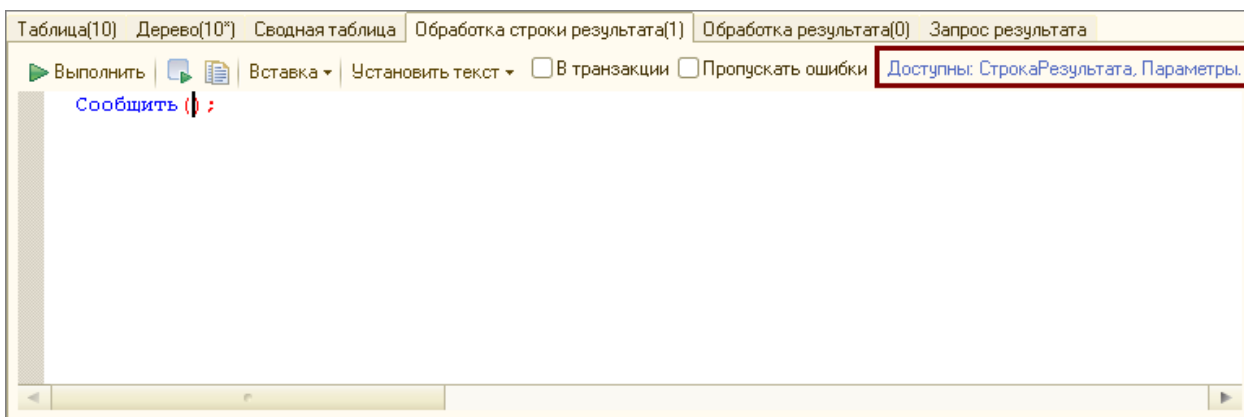
Итак, мы выбрали нужную нам номенклатуру и хотим ее обработать. Например, мы можем быстренько написать какой-то небольшой код и что-то сделать с этими строчками номенклатуры. Допустим, сейчас для демонстрации того, как это можно сделать, мы просто Сообщить (Наименование) сделаем по ссылке.

Здесь у нас есть два обработчика строки – обработка строки результата и обработка результата. Чем они собственно отличаются?

- Обработка результата в качестве параметра здесь на вход получает объект типа «Результат запроса». Вы сами можете из него выгрузить данные, в таком виде, в каком вам нужно и обойти их этими выборками или выгрузить в дерево и там их как-то обойти.
- А обработка строки результата – тело цикла обработки таблицы результата Для каждого Строка Из ТаблицаРезультата Цикл... (ТаблицаРезультата – это та таблица, которая у нас на закладке «Таблица» расположена) и вот дальше в цикле идет то, что здесь у нас будет написано.

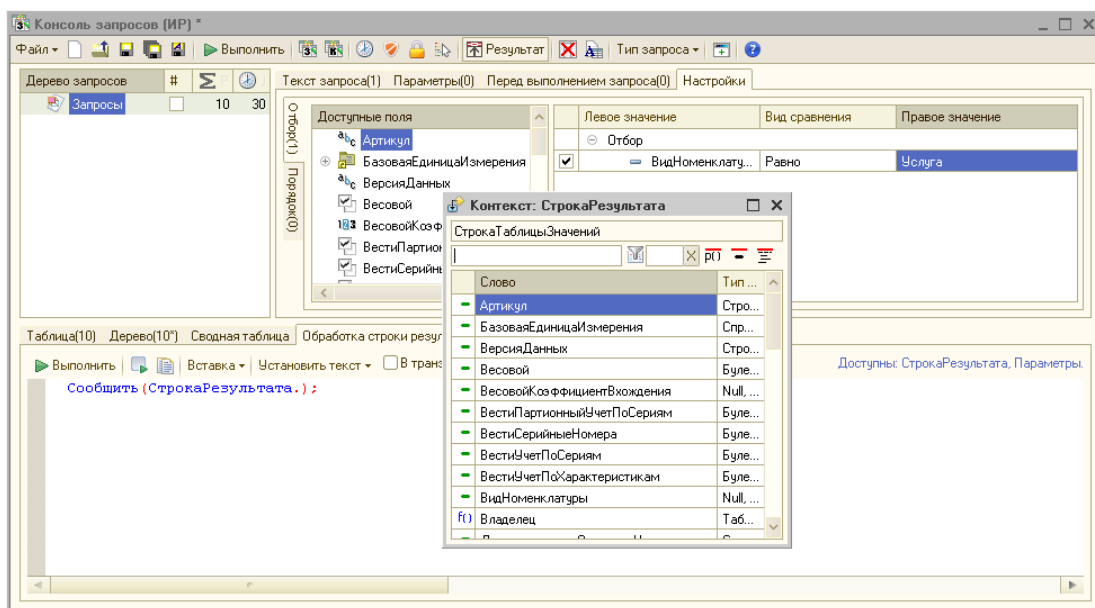
Мы переходим на закладку «Обработка строки результата», пишем там оператор Сообщить() – и дальше обращаемся к строке этой таблицы.

Обратите внимание, здесь есть пояснение: «Доступны: СтрокаРезультата, Параметры».



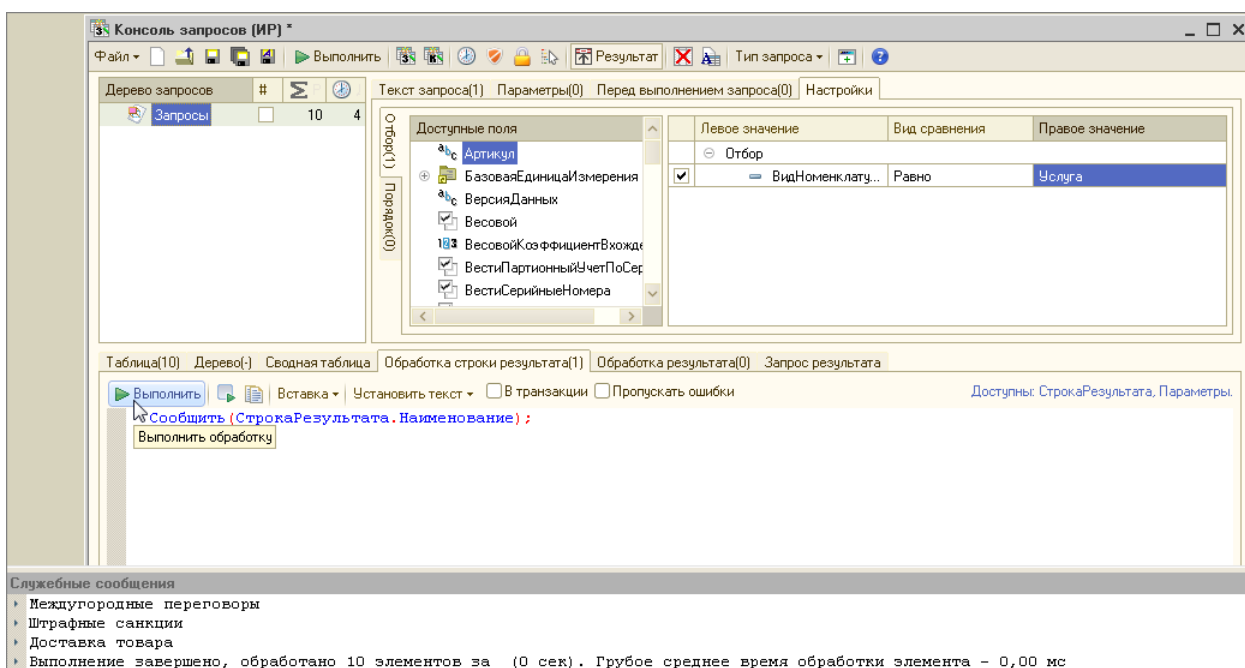
Параметры – это структура, в которой содержится значение всех параметров запроса. Если вам надо использовать в этом коде какие-то дополнительные ссылки на объекты базы данных (например, значение каких-то реквизитов), то вы можете их описать в виде значений параметров запроса и использовать в этом коде.

Вот я обращаюсь к строке результата, ставлю точку и вызываю контекстную подсказку Ctrl+Пробел.



Здесь мне подсказываются все реквизиты. Обратите внимание, что мы сейчас видим именно колонки таблицы нашего результата. Наверху написано, что это СтрокаТаблицыЗначений.

И мы здесь, например, наименование выведем.



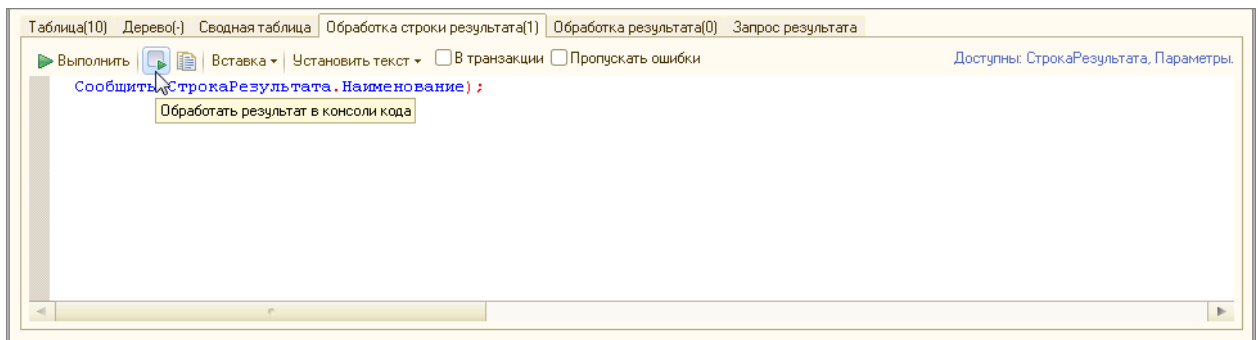
Я нажимаю «Выполнить» и вывожу на экран все наименования.

Обработка результата в консоли кода

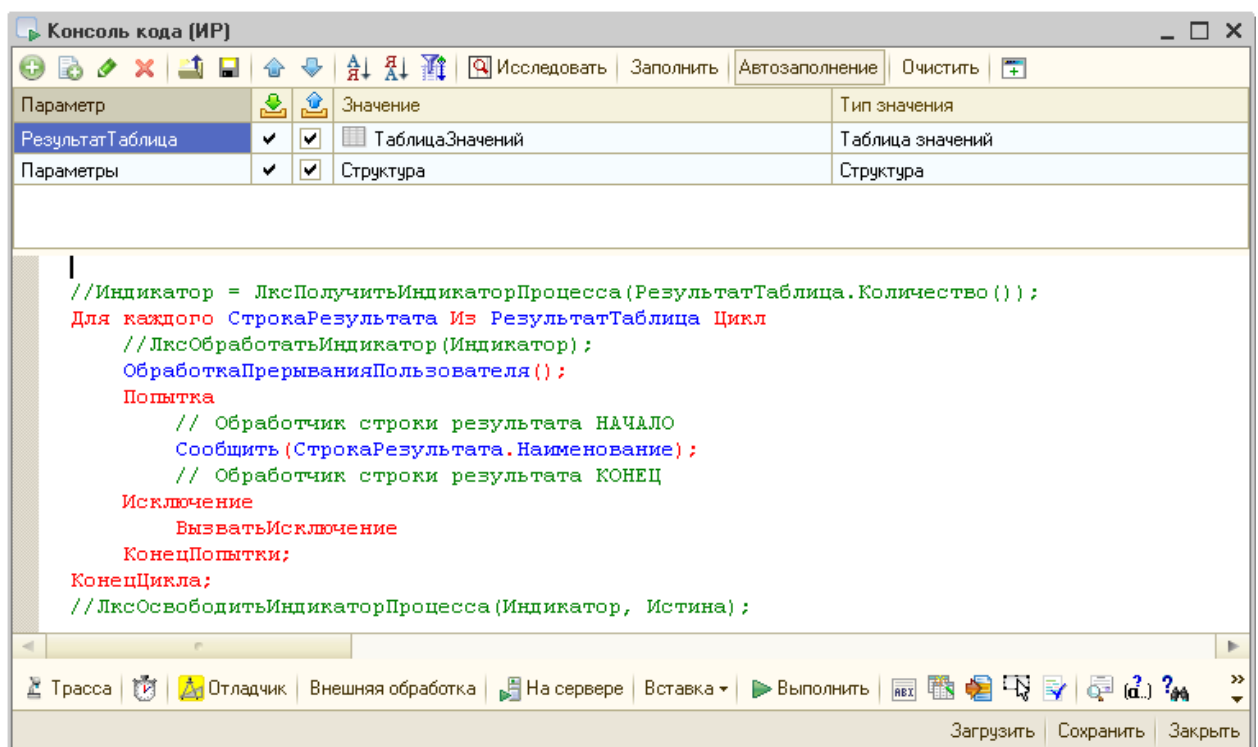
Как отладить теперь этот код?

Сейчас у нас в консоли запросов написан элементарный код. Можно сказать, здесь даже отлаживать-то нечего. Но если вы войдете во вкус, то сначала конечно что-то простое

напишете, но напустит момент, и вы напишете достаточно сложный код, который захочется отладить. Это можно сделать с помощью кнопки «Обработать результат в консоли кода».



Давайте нажмем на нее. Что здесь происходит?



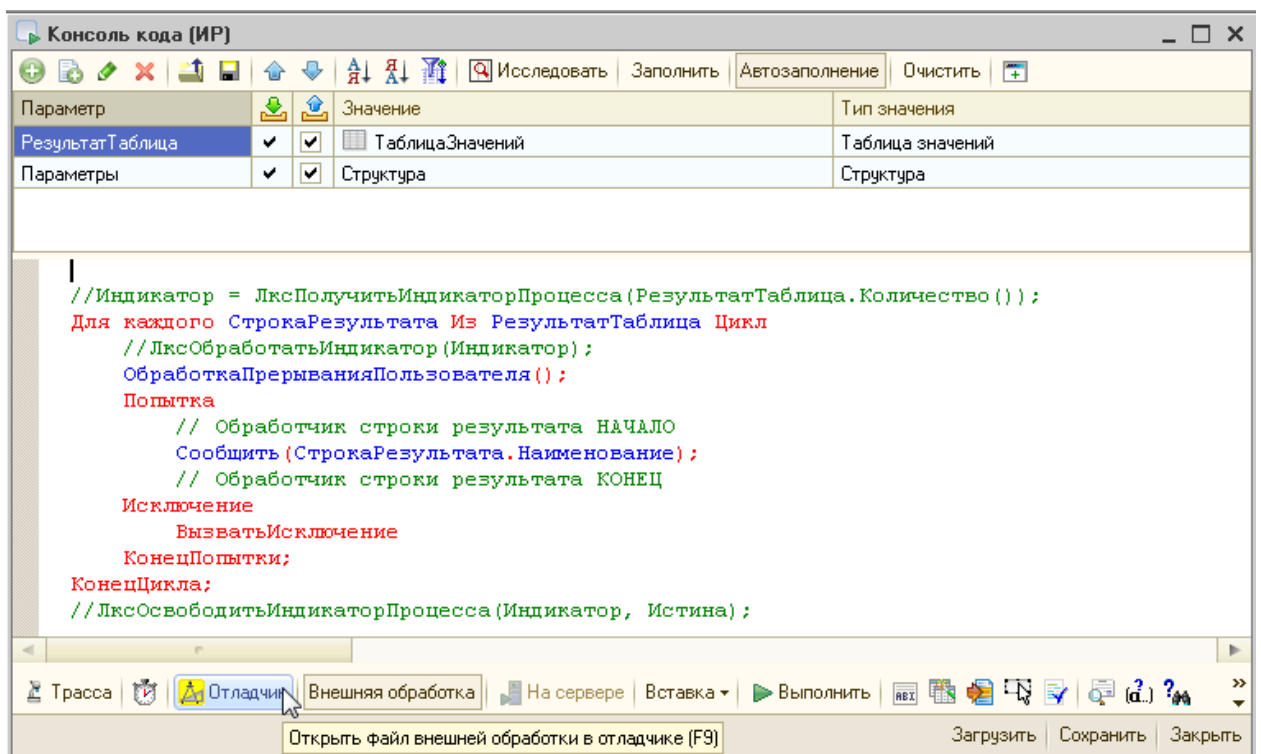
При нажатии на эту кнопку генерируется тело цикла для обхода всех строк таблицы результата, в который передаются параметры (через структуру всех значений параметров) и сама таблица результата.

В теле цикла у нас собственно тот самый код, который мы писали в том поле текстового документа. Он обрамлен комментариями, чтобы было понятно, что это именно он.

Здесь при нажатии кнопки «*Выполнить*» мы получаем тот же самый эффект – выводятся все наименования.

Отладка результата в конфигураторе

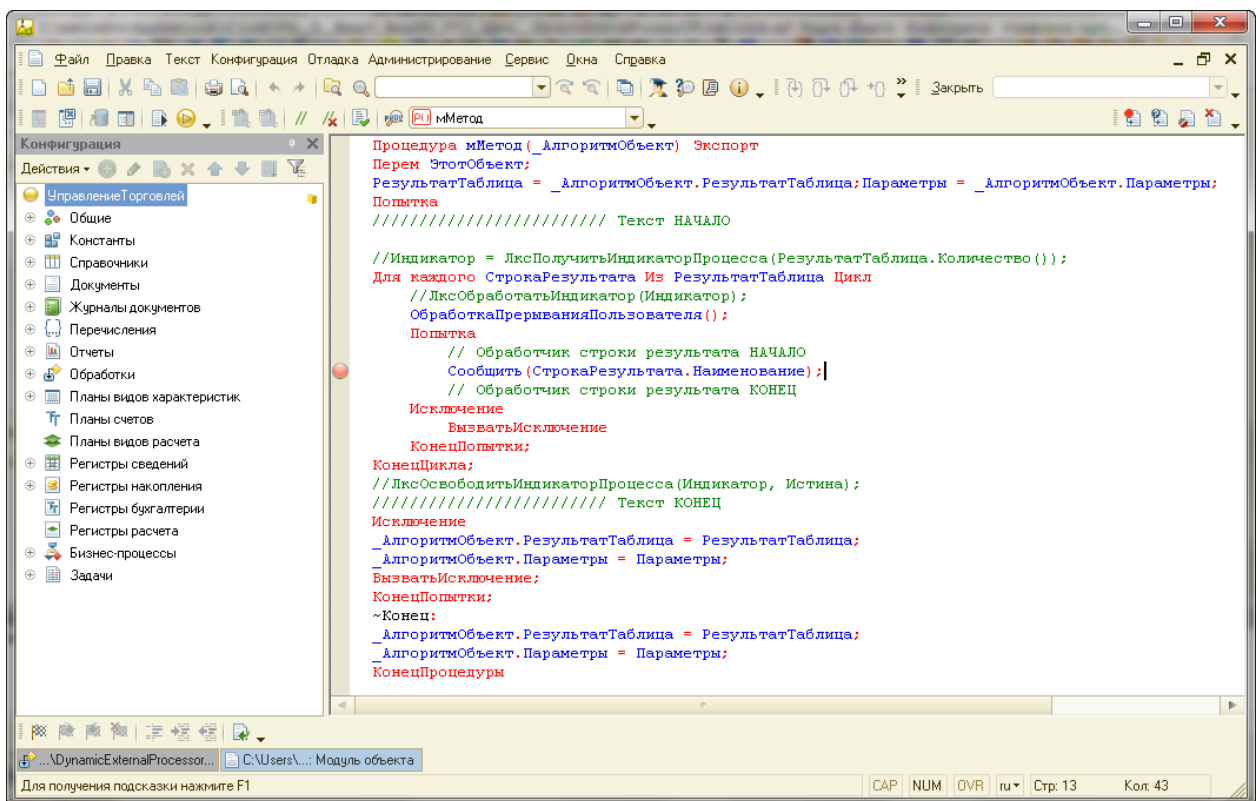
Как теперь этот код отладить в конфигураторе?



Для этого мы в консоли кода включаем режим «Внешняя обработка» и нажимаем кнопку «Отладчик».

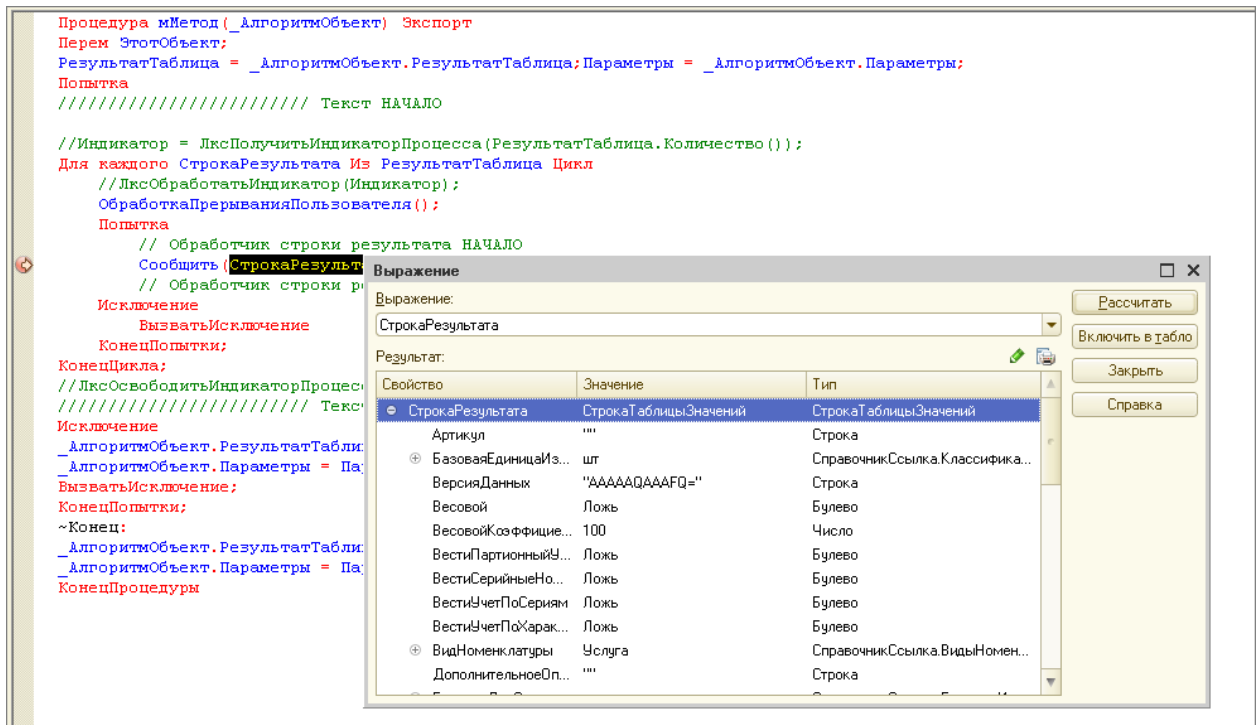
У нас открывается динамически сформированная внешняя обработка, в которую подсунут уже специальный модуль, где мы видим наш код из консоли кода.

Здесь мы можем поставить точку останова. К примеру, на строке с методом Сообщить(). Допустим, у нас там что-то не то происходит или нам так кажется.



Переключаемся в режим Предприятие.

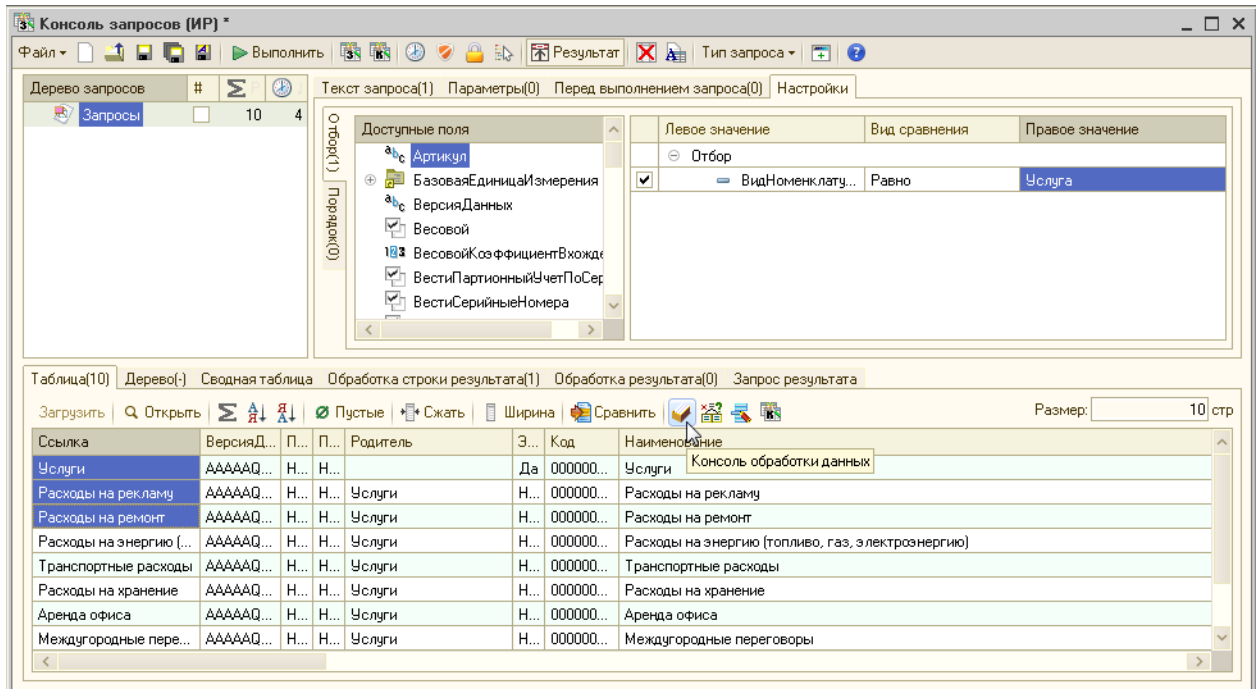
Нажимаем «Выполнить». Видим, что у нас отладчик остановил выполнение программы.



Стоя на строке мы можем вычислить выражение, т.к. все стандартные средства нам здесь доступны. Фактически, это – предоставление всех удобств отладчика для работы с динамическим кодом.

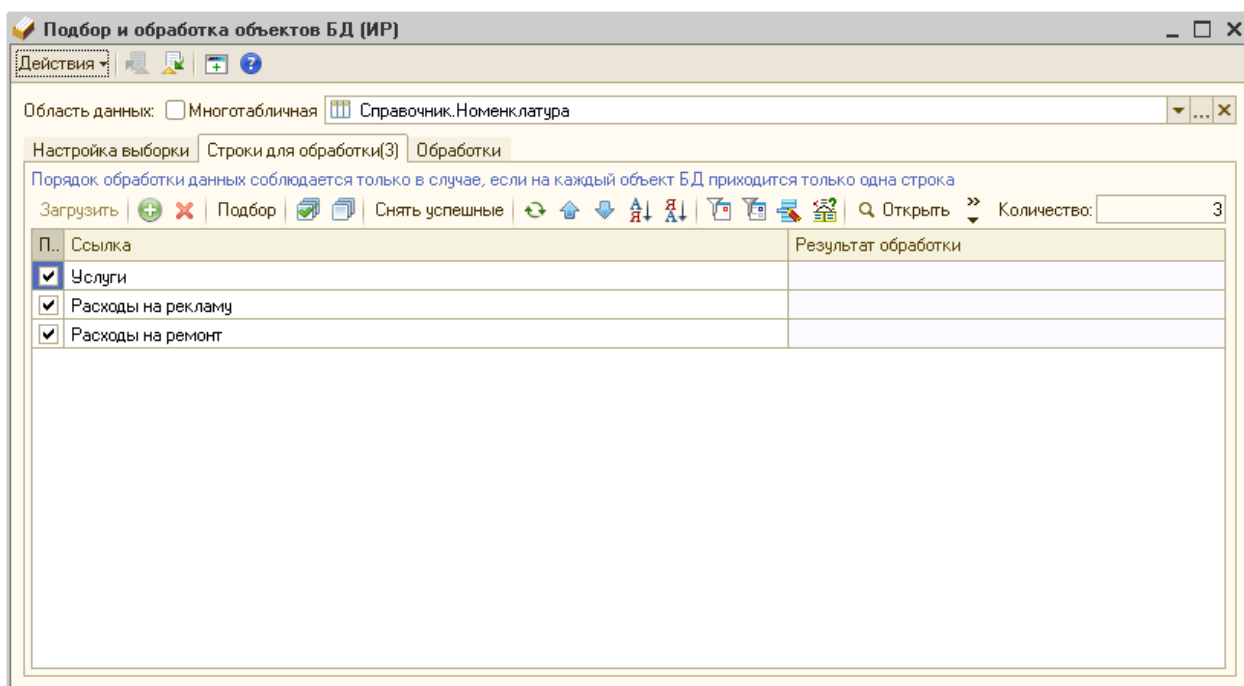
Передача части результата запроса в инструмент «Подбор и обработка объектов»

Кроме этого, из консоли запросов мы можем вызывать другие инструменты подсистемы «Инструменты разработчика».



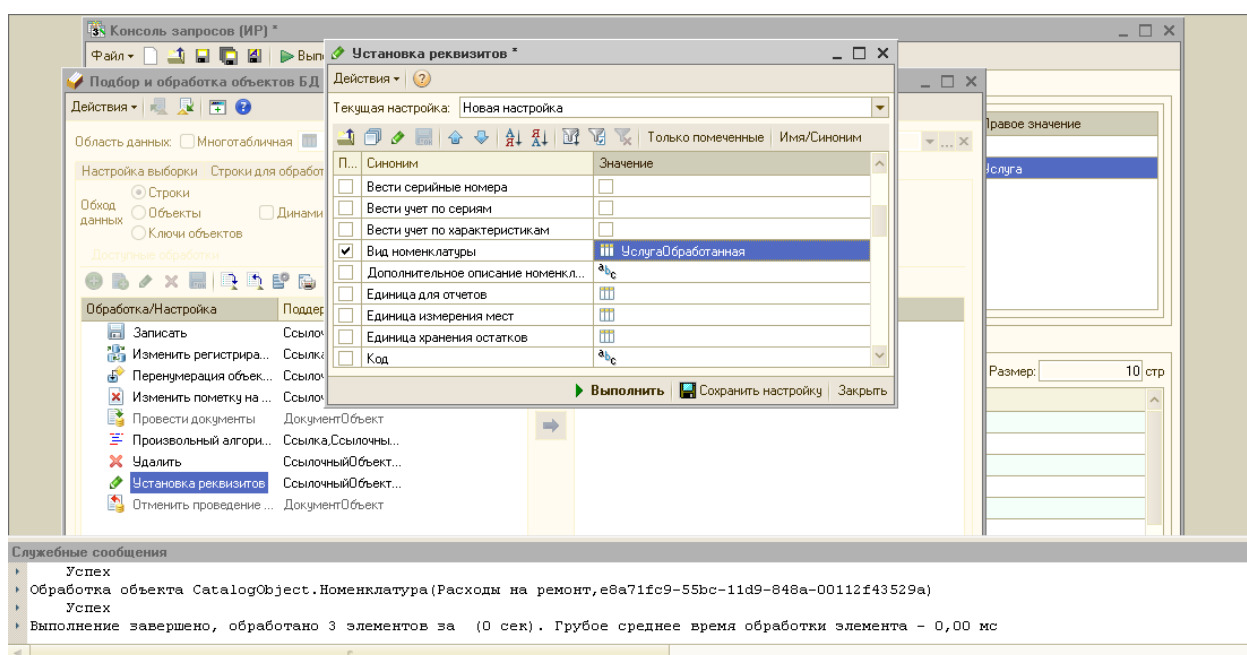
Допустим, мы с помощью множественного выделения в таблице результата выделили какие-то нужные нам три ссылки и вызываем инструмент «Консоль обработки данных». Здесь эта команда называется немножко по-старому, на самом деле это «Передача выделенных ссылок в инструмент «Подбор и обработка объектов».

У нас открывается инструмент «Подбор и обработка объектов». И в нем уже заполнены строки для обработки.



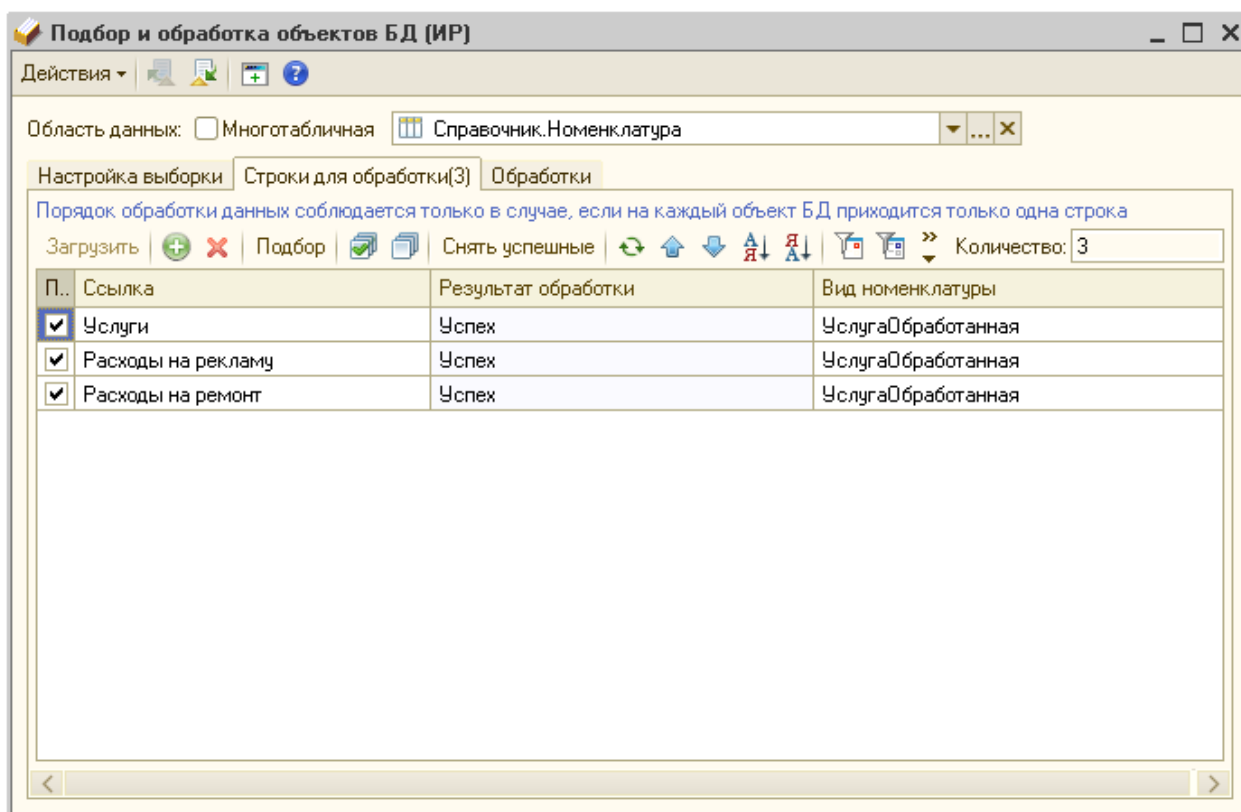
Здесь мы дополнительно можем какой-то флажок снять, если вдруг мы решили, что нам его не надо обрабатывать.

На закладке «Обработка» выбираем какую-нибудь обработку. Например – установка реквизитов.



И меняем вид номенклатуры для отобранных позиций. Нажимаем кнопку «Выполнить» и переключаемся обратно на закладку «Строки для обработки».

Обратите внимание, что тот реквизит, значение которого мы изменяли вот этой встроенной обработкой, автоматически добавил сюда свою колонку – и мы увидели новое значение этого реквизита и результат обработки – Успех.

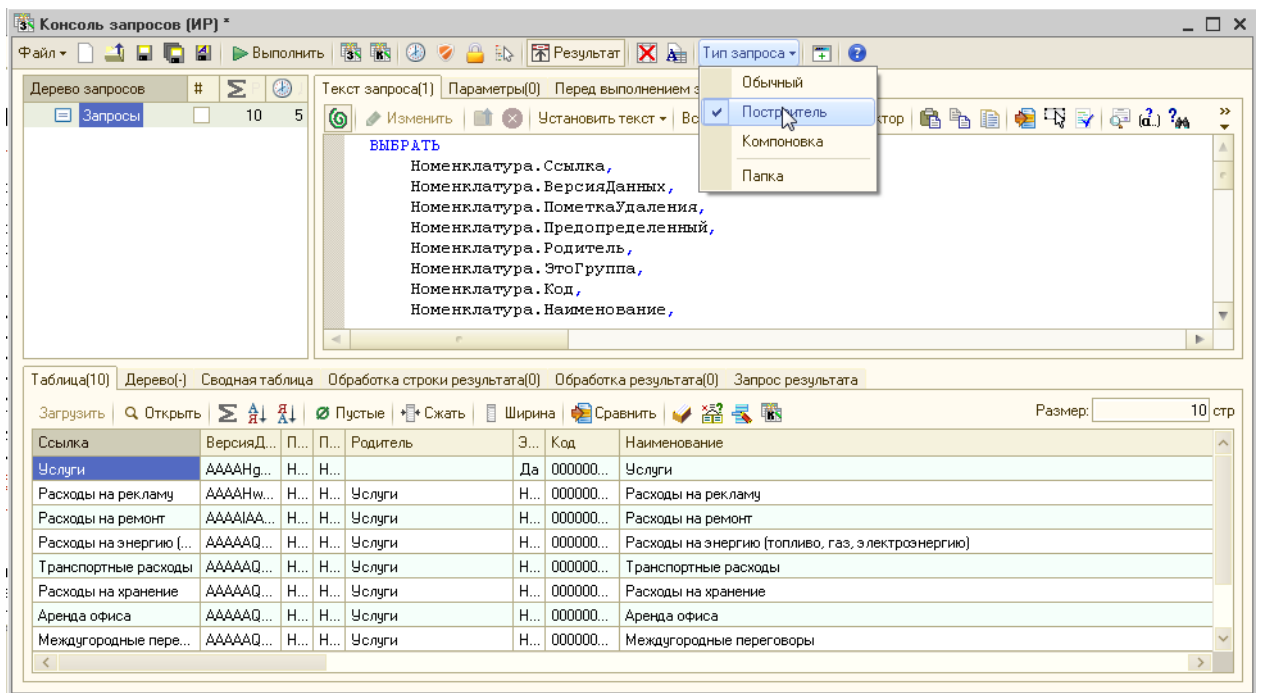


То есть, если бы здесь какой-то объект не обработался (например, ошибка перед записью бы возникла), то здесь бы было описание ошибки и в виде номенклатуры осталось бы старое значение, естественно.

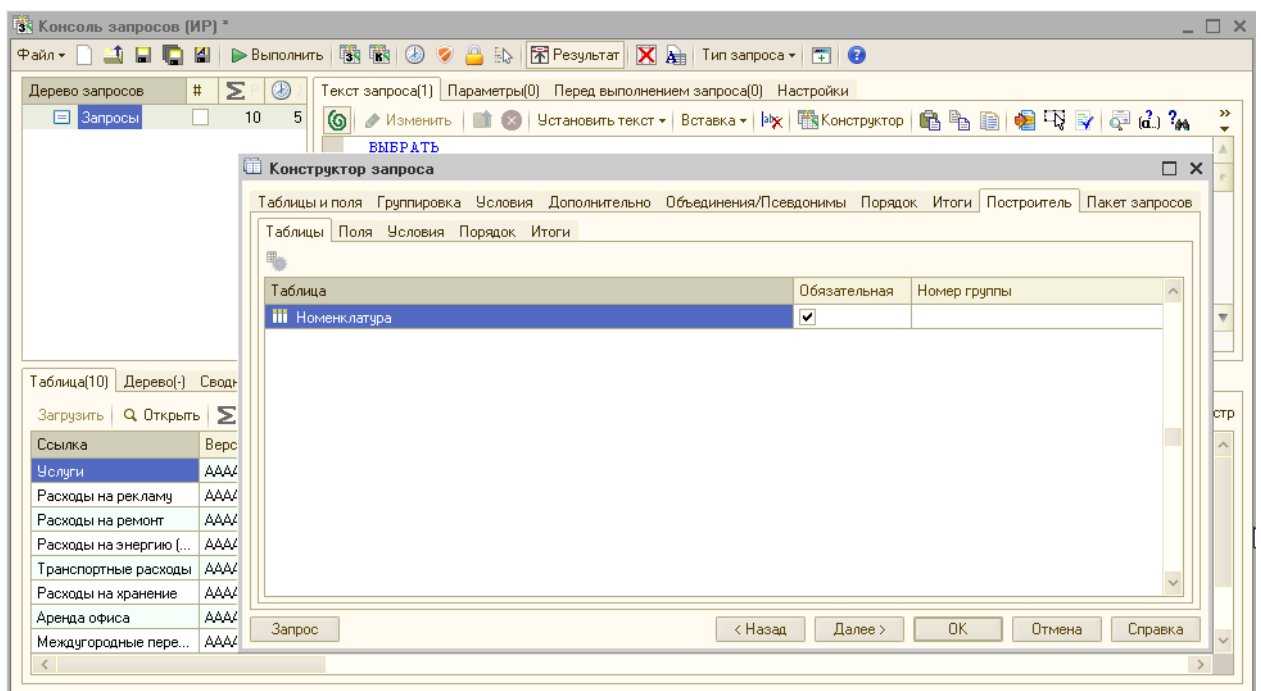
Вернемся к консоли запросов.

Работа с консолью запросов в режиме «Построитель»

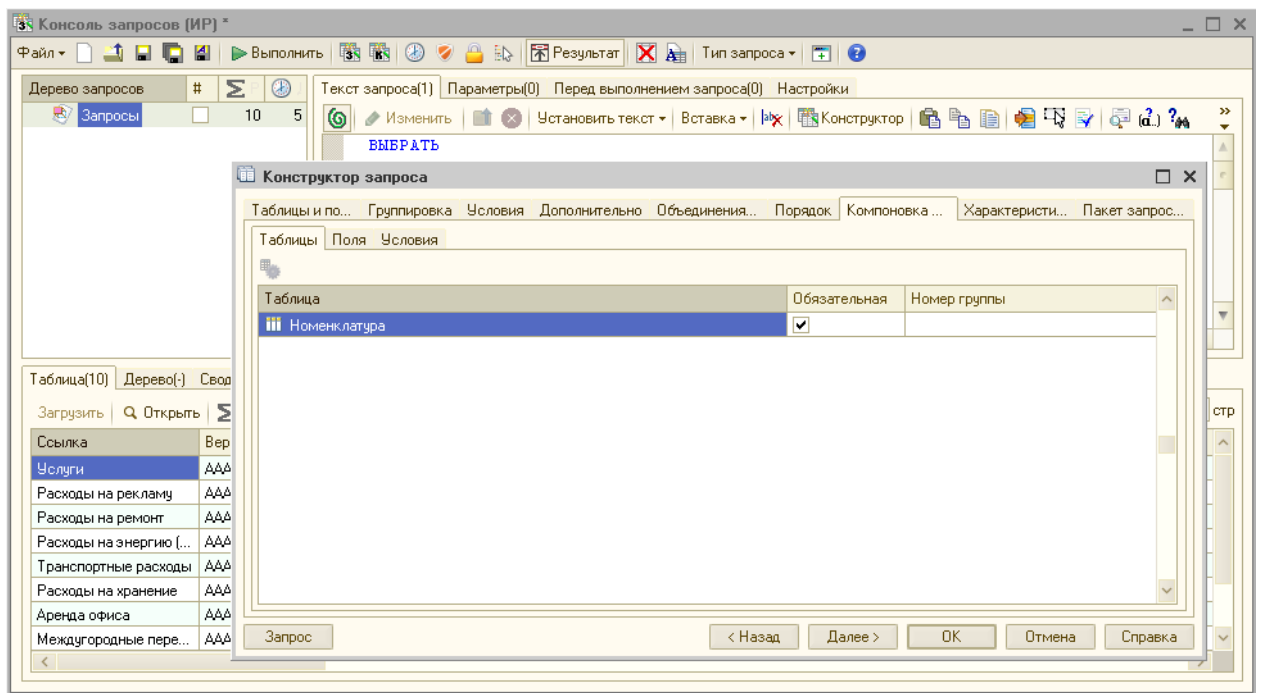
В консоли запросов есть режим построителя. Наверное, в последних конфигурациях он уже мало где используется. Но я уверен, есть много случаев, когда он нам все-таки может понадобиться.



Обратите внимание, когда мы переключаем тип запроса на построитель, при вызове конструктора запроса появляется закладка «Построитель».



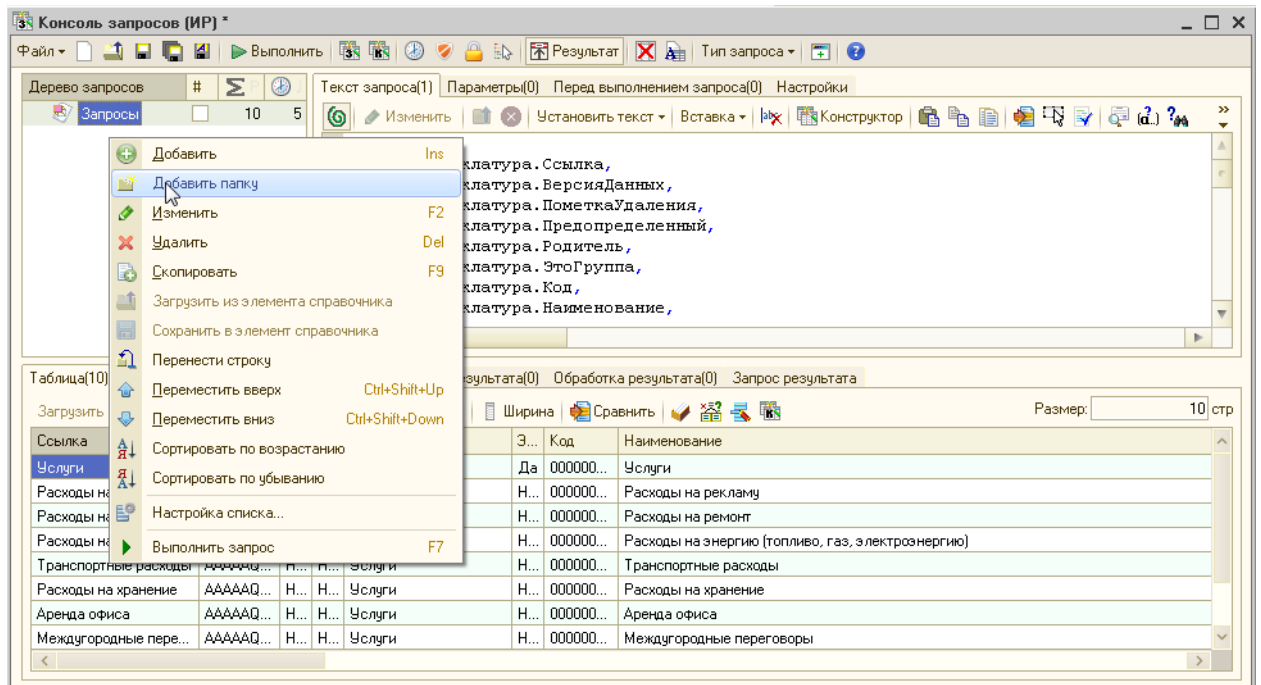
А когда переключаем обратно на компоновку, здесь появляется закладка «Компоновка»



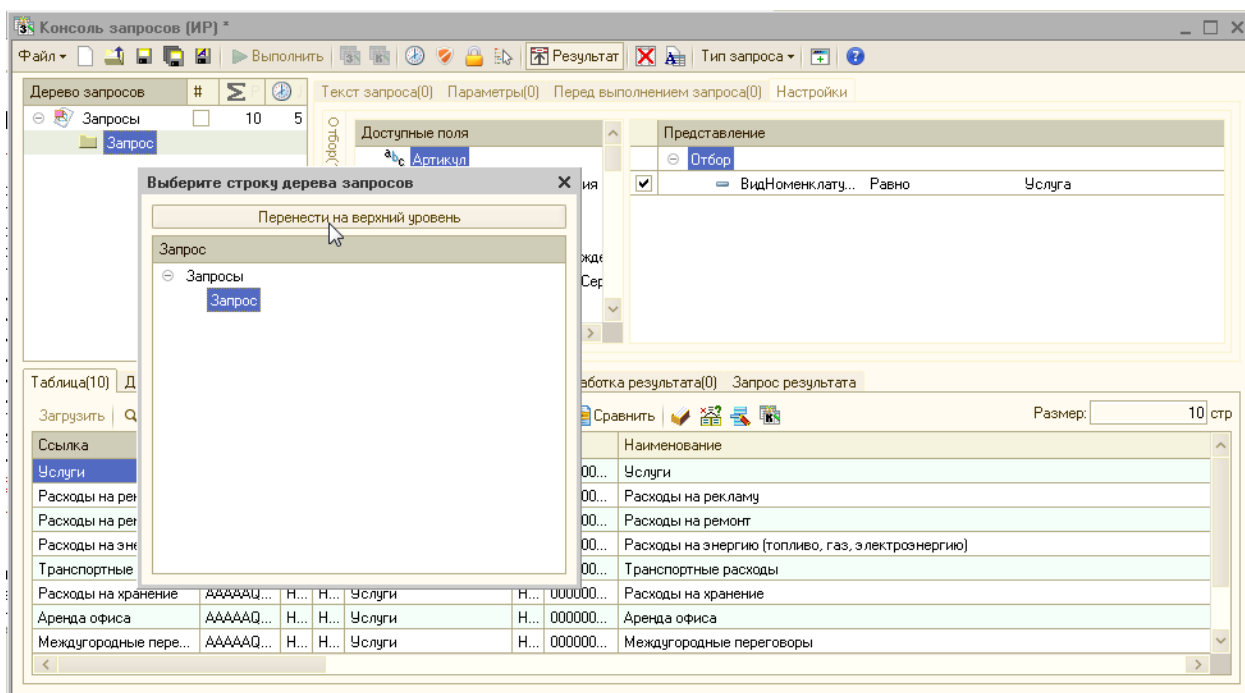
То есть, все особенности поведения полностью переключаются в соответствии с типом запроса.

Иерархия в дереве запросов

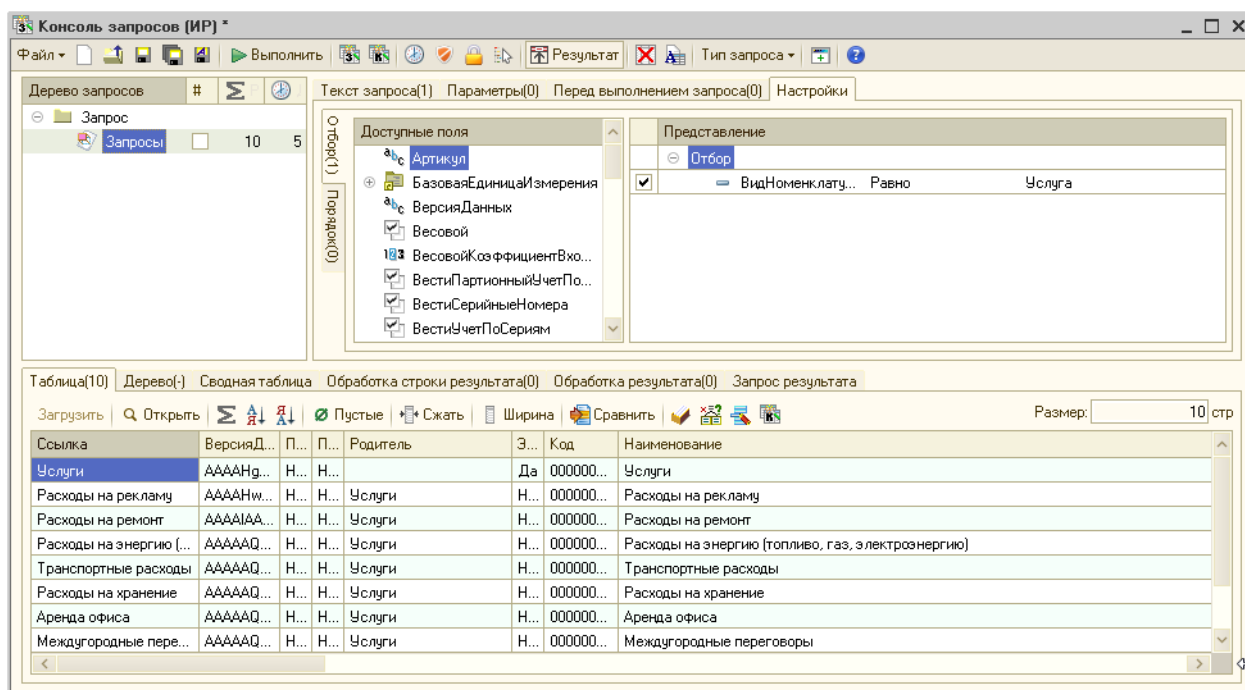
Зачем нужна папка в дереве запросов?



Папка служит для группировки строк дерева и при ее активизации становятся недоступными большинство элементов управления запросом. Ее можно создавать в любом месте дерева.

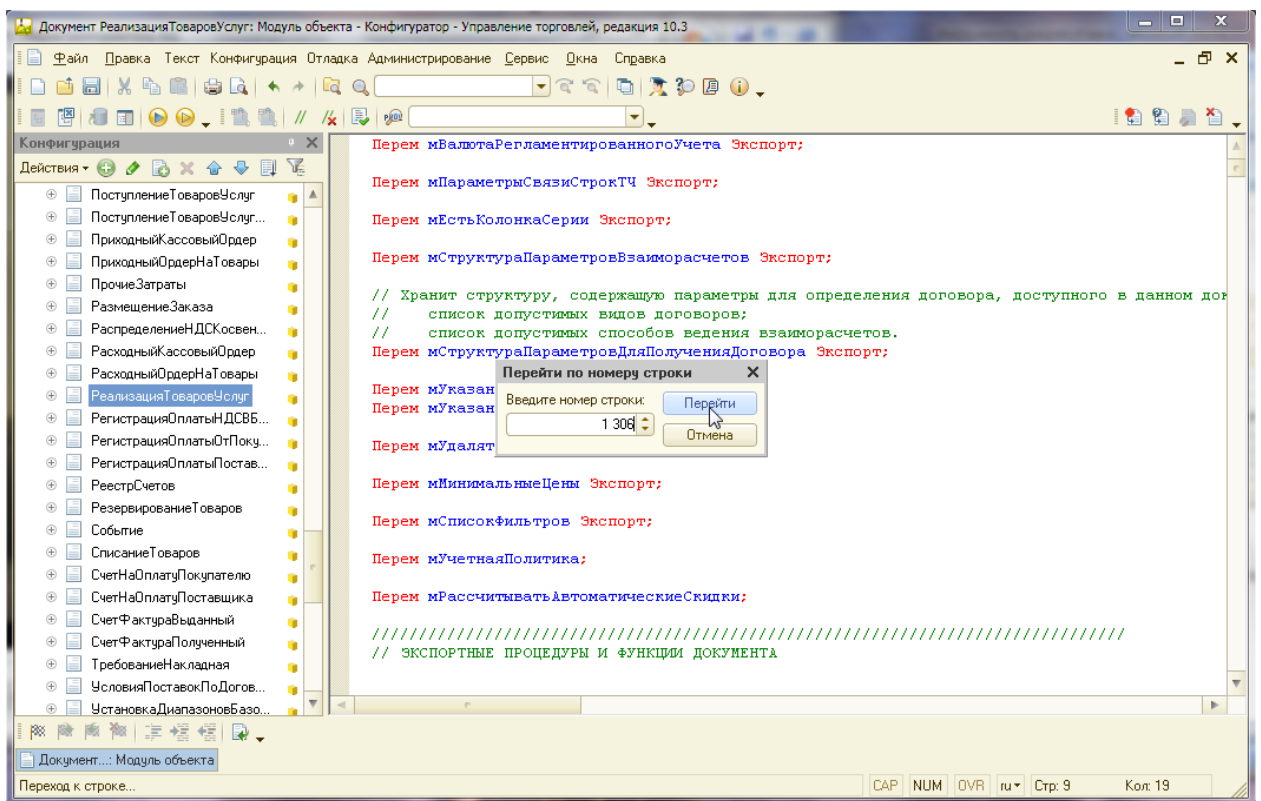


Строку дерева запросов можно переместить в любую ветку, исключая ее потомков.

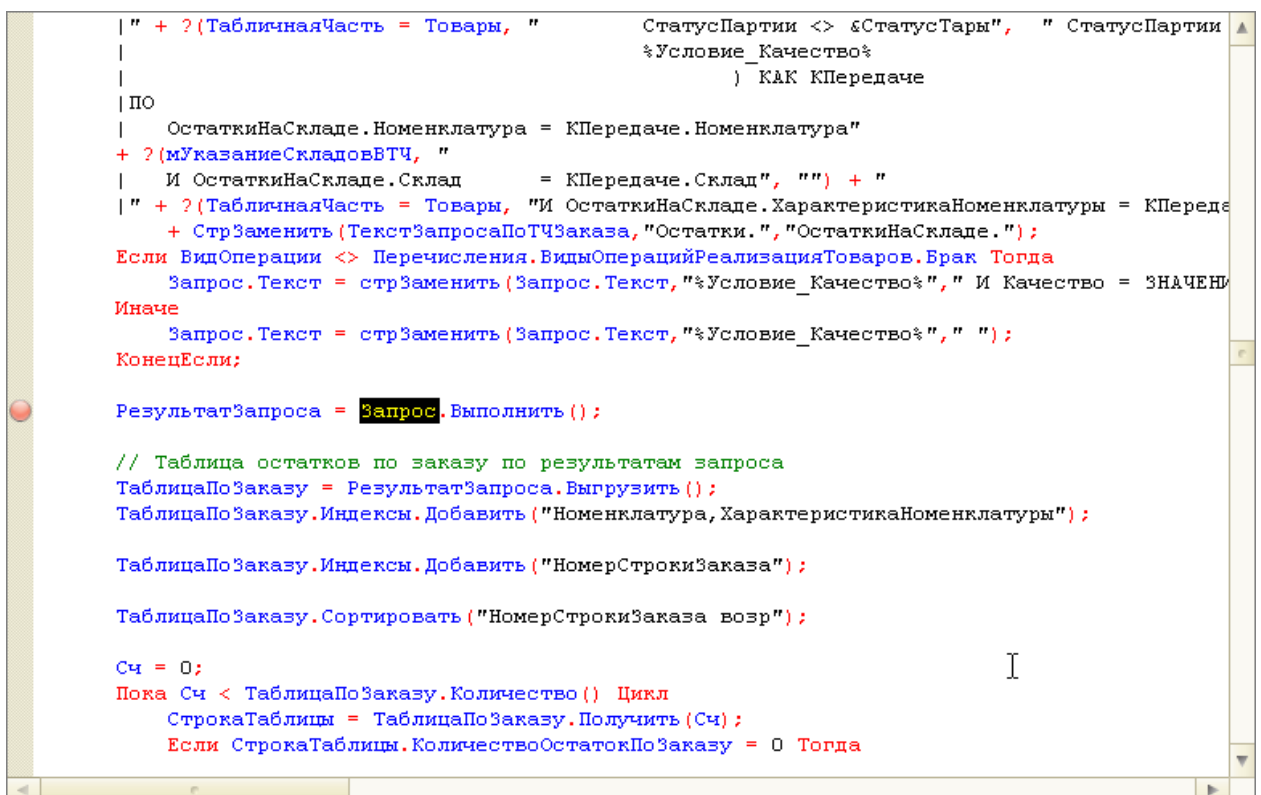


Работа с консолью запросов в режиме отладки

Теперь посмотрим на работу в консоли запросов с деревом запроса. Лучше всего это сделать на примере какого-то реального запроса.



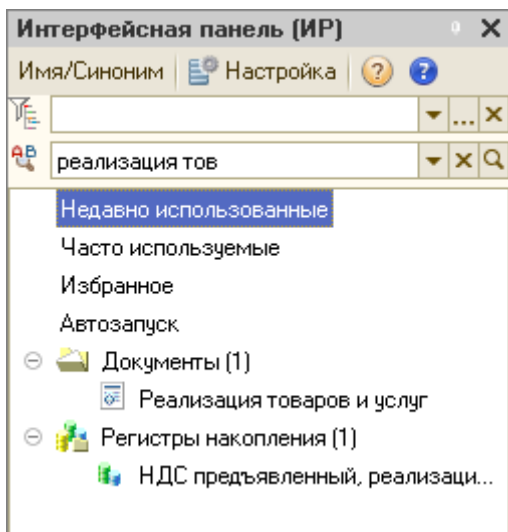
Для этого мы перейдем в конфигуратор и откроем там документ «Реализация товаров и услуг» на строке модуля 1306.



Здесь у нас запрос, который заполняет табличную часть документа. Он собирается динамически – видите, тут есть СтрЗаменить и много разных условий.

Поставим здесь точку останова.

А теперь откроем в режиме предприятия расходные накладные. Заодно я продемонстрирую работу инструмента «Интерфейсная панель».



Здесь я могу ввести два слова и обработка отберет мне те объекты, которые эти слова содержат.

В списке реализаций выберу документ, введенный на основании какого-нибудь заказа покупателя – открываем его, очищаем табличную часть и вызываем действие «Заполнить по заказу покупателя»

Реализация товаров и услуг: продажа, комиссия. Проведен *

Операция ▾ Цены и валюта... Действия ▾

Номер: от: Отразить в: ☒ упр. учете ☒ бух. учете ☒ налог. учете

Организация: Со склада

Контрагент: Договор:

Заказ покупателя: По договору с покупателем нет долга

Товары (0 поз.) Тара (0 поз.) Услуги (0 поз.) Дополнительно Автоматические скидки Печать Предоплата

Заполнить ▾ Подбор Изменить **N2** Серийные номера

Заполнить по заказу покупателя (счету на оплату)

Отбирать заказы, заполнить по заказам

Добавить из поступления

1 USD = 25,9839руб., Тип цен: Оптовая, Условие продаж: Безналичная оплата Всего (USD):

Счет-фактура: НДС (сверху):

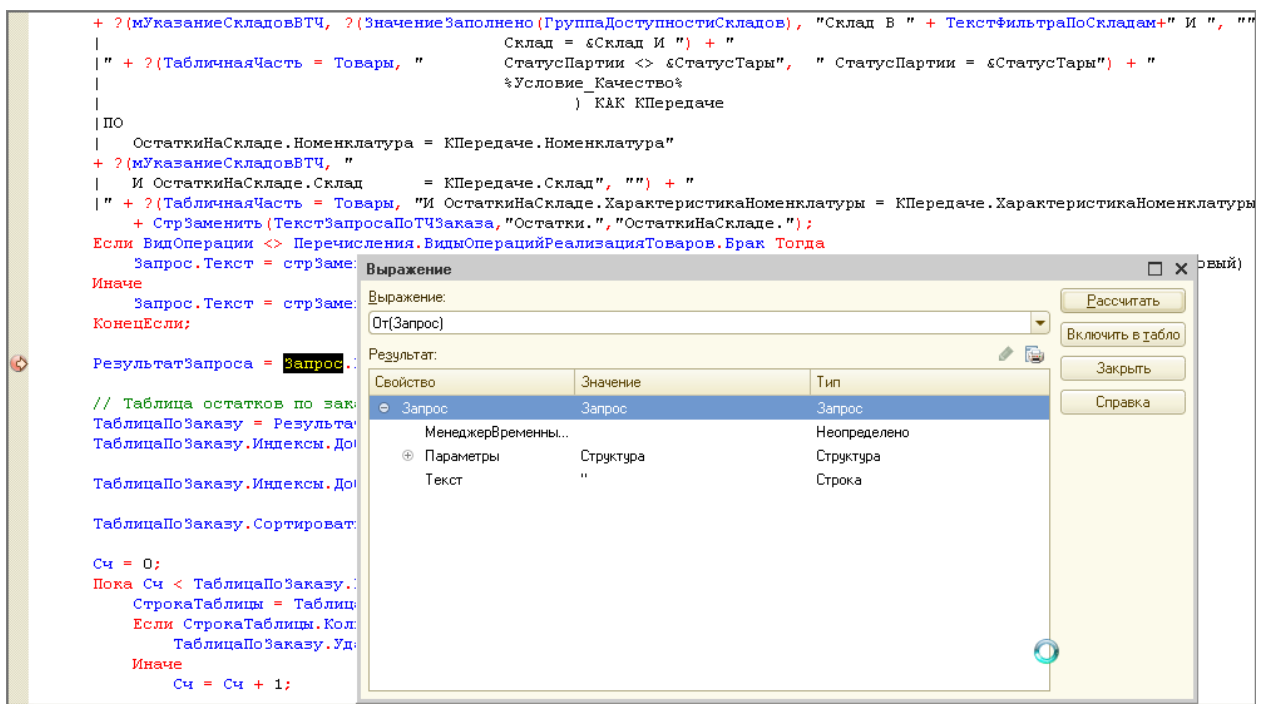
Комментарий:

Расходная накладная (вал.) **OK** Записать Закреть

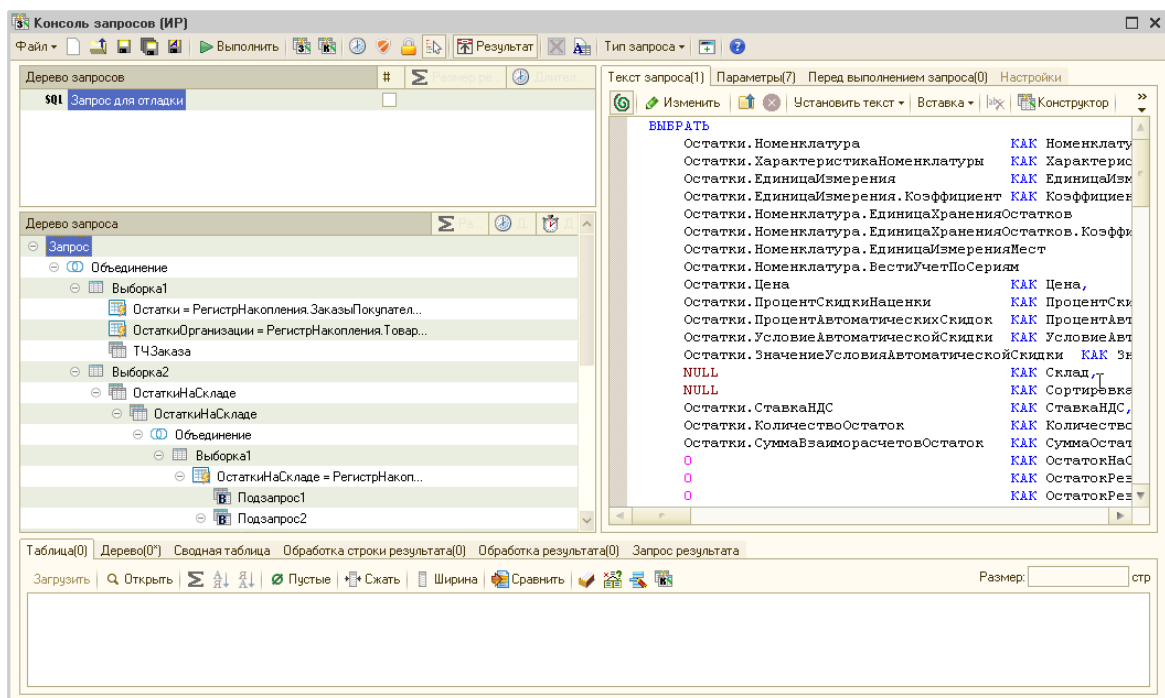
Работа с функцией режима отладки «От»

Заодно продемонстрирую работу функции режима отладки «От».

В точке останова на выделенной переменной Запрос я вызываю диалог «ВычислитьВыражение» Пишу в нем От(Запрос) и нажимаю «Рассчитать».



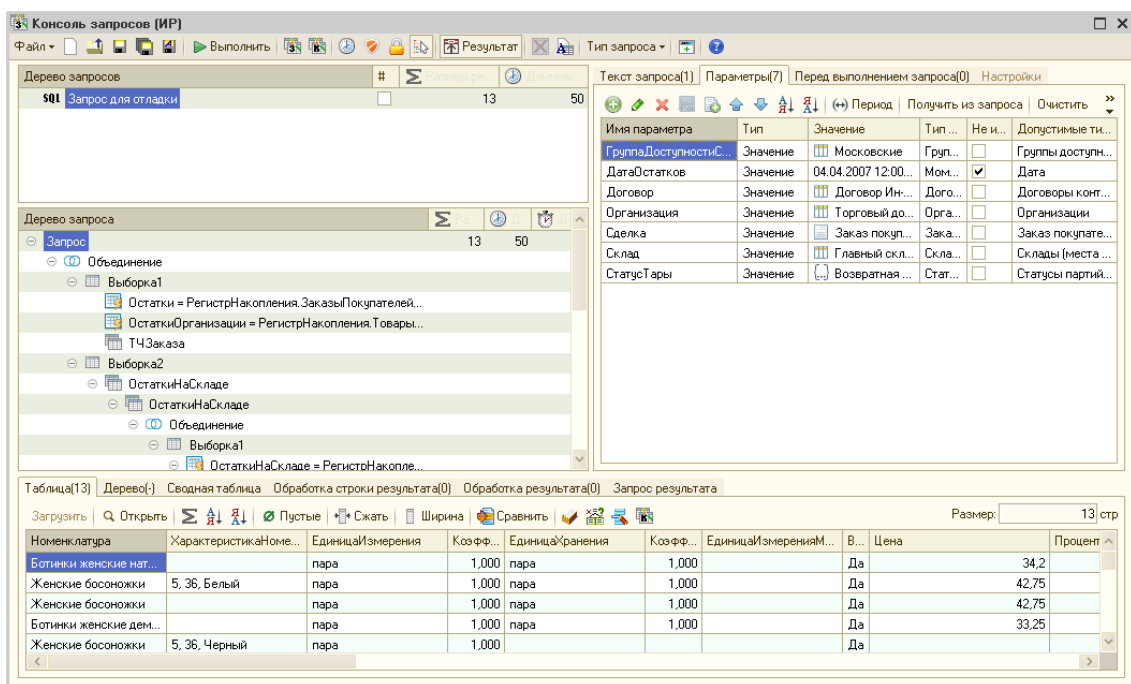
Приложение отладчика переходит в режим ожидания (оно показывает «Часики»). А в это время в режиме предприятия у меня открывается консоль запросов, в которой мы можем сразу увидеть наш запрос. Причем для него сразу построено дерево запроса.



Работа в консоли запросов с деревом запроса

Что это за дерево запроса? Мы видим здесь, что в корне запроса у нас объединение, внутри этого объединения две выборки объединяются. Посмотрим на параметры. Видим, что параметры все заполнены. То есть, запрос полностью работоспособный.

Нажимаем «Выполнить» и получаем табличную часть результата.



Заполнилось 13 строк. Сейчас мы акцентируемся на дереве запроса.

Итак, мы видим здесь несколько колоночек.

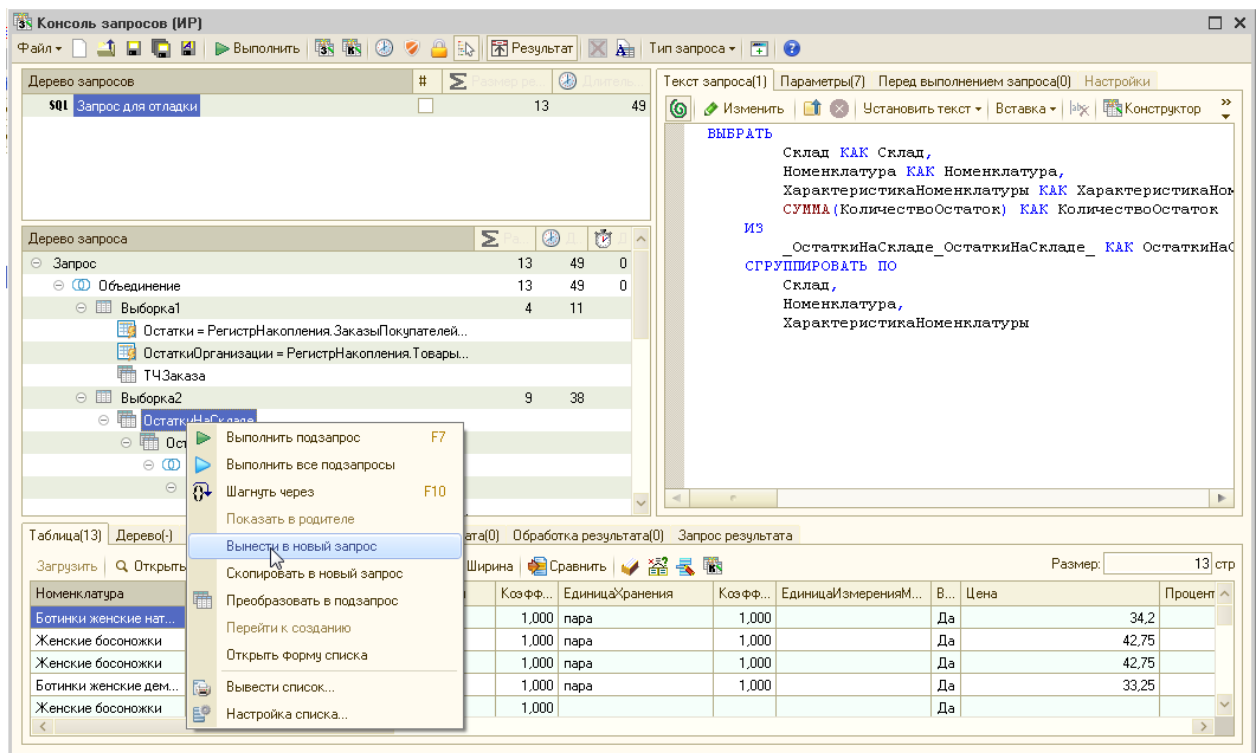
- Первая – это **число строк результата последнего выполнения**. Эта колонка нужна, чтобы видеть, например, изменилось ли число строк результата, когда вы заново выполняете после редактирования. Это самый простой критерий, как понять, что вы изменили логику запроса. Допустим, вы не хотели ее менять, но поменяли что-то, организацию в доступные поля добавили и потом – раз, смотрите, а число строк изменилось. Если это какая-то живая база рабочая, то это способ подстраховаться в какой-то мере от нежелательных изменений.
- Дальше – колонка **«Длительность выполнения»**. Она для каждой веточки вычисляется, если вы по ней два раза кликнете. Вот, например, я сейчас стою на объединении – в данном случае длительность совпадает с основным запросом.

Здесь в подсказке написано, что она вычисляется, как полная сумма для текущей строки (как его полная длительность минус разница длительностей дочерних веточек). Грубо говоря, это позволяет оценить, на каком уровне в дереве находятся какие-то проблемные места в запросе, где лучше начинать поиск места для оптимизации. По-взрослому, конечно, надо посмотреть на план запроса, но для этого какое-то время требуется и не всегда есть возможность его посмотреть. И, к тому же, получить в плане запроса красивое дерево можно только для плана запроса MS SQL. Для плана запроса файловой СУБД никакой красивой картинки не будет, будут просто текстовые блоки.

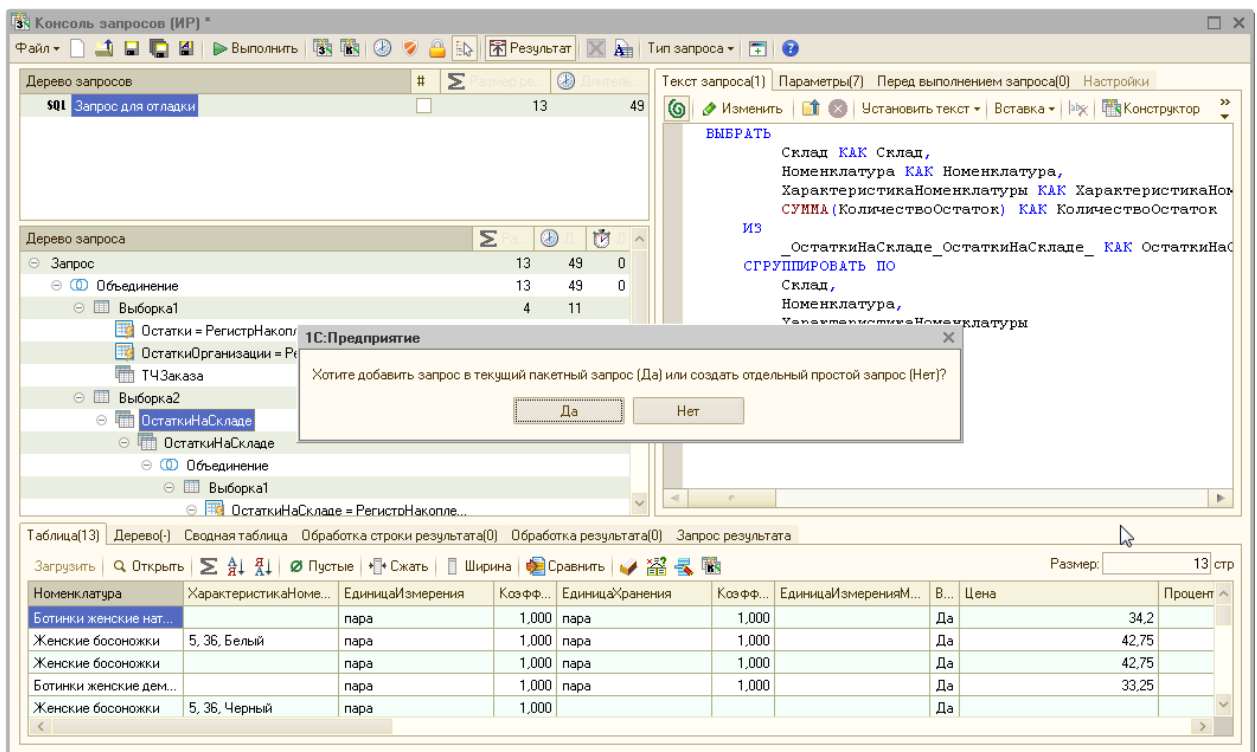
Вынос подзапроса во временную таблицу

Что еще в дереве запросов можно делать? Как я и говорил, здесь есть еще полезные функции – например, мы почему-то решили этот подзапрос вынести во временную таблицу, нам показалось, что это скажется положительно на времени выполнения этого запроса в целом. Для оптимизации.

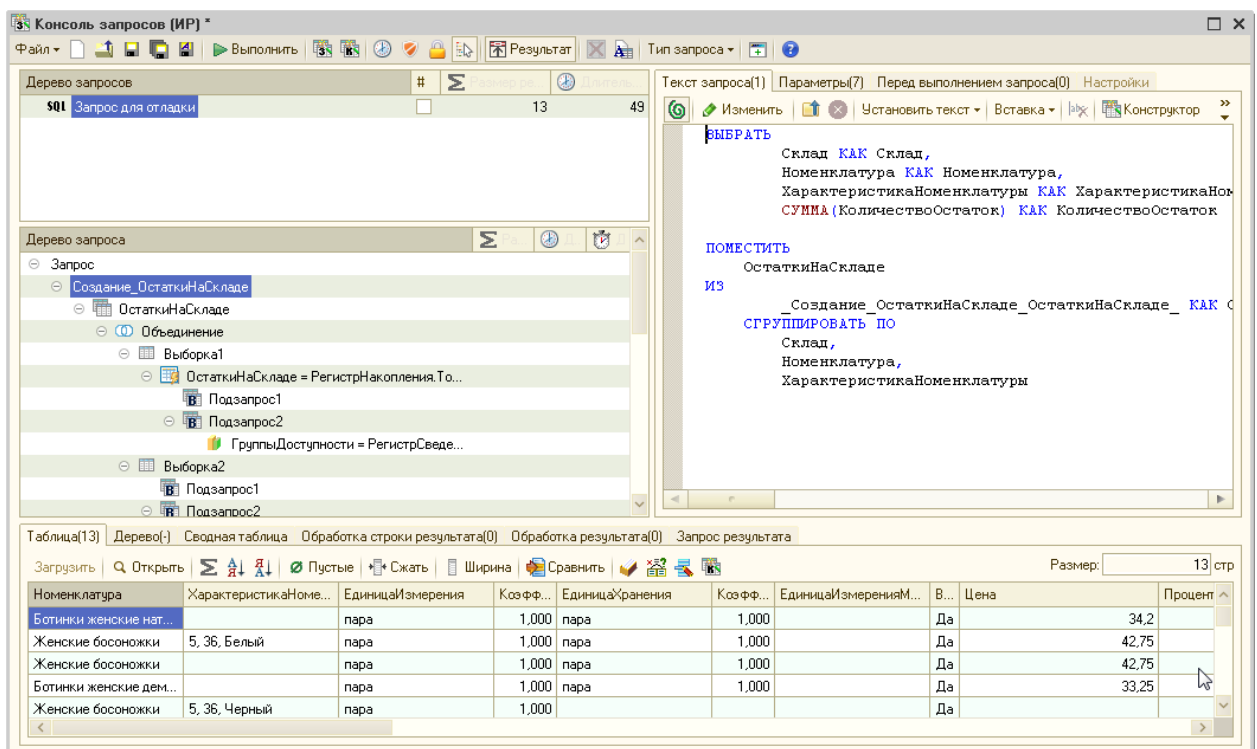
Вызываем контекстное меню, где выбираем пункт «Вынести в новый запрос»



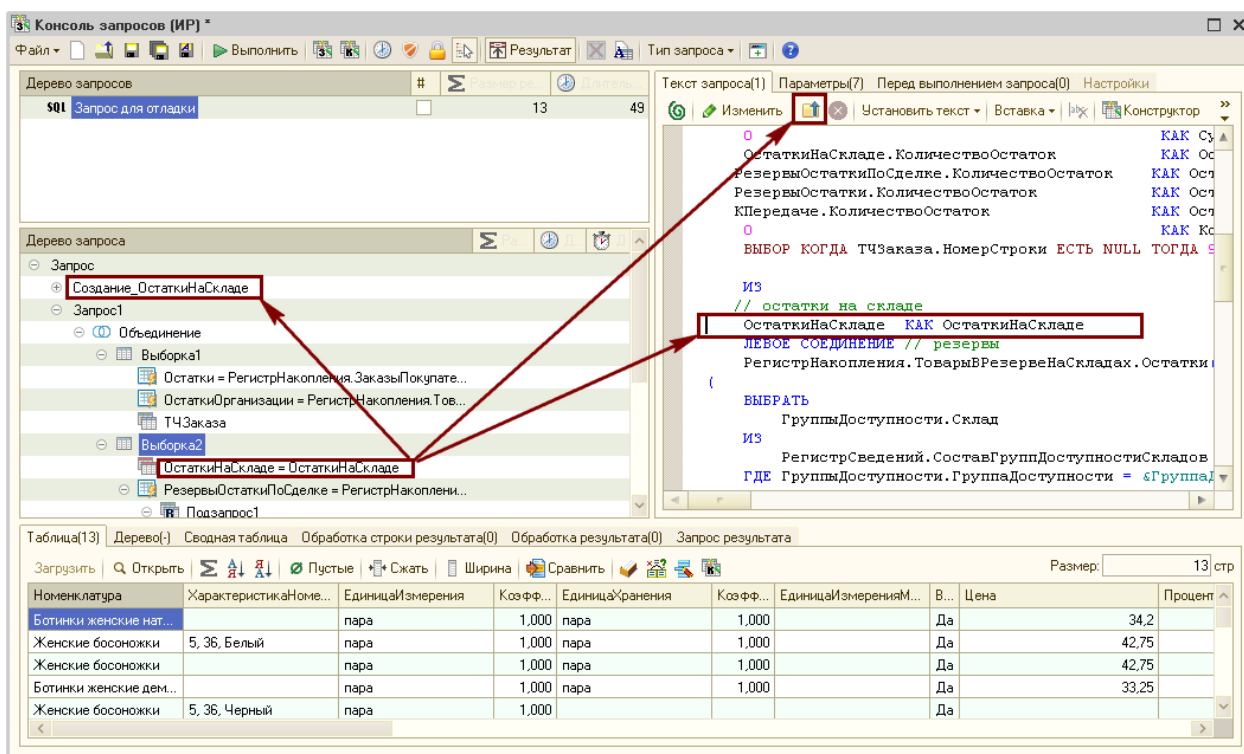
Здесь у нас есть возможность либо в отдельный запрос в дереве запросов (добавить новый элемент) либо в текущий пакетный запрос (самый классический случай) – добавить новый запрос, в который будет помещен вот этот самый запрос, который находится в этом месте дерева.



Нажимаем «Да». Что мы получаем? Как мы видим, в дереве появился новый элемент. Все запросы на создание временных таблиц имеют имя Создание_ и дальше имя временной таблицы, чтобы их легче было отличать от обычных запросов.



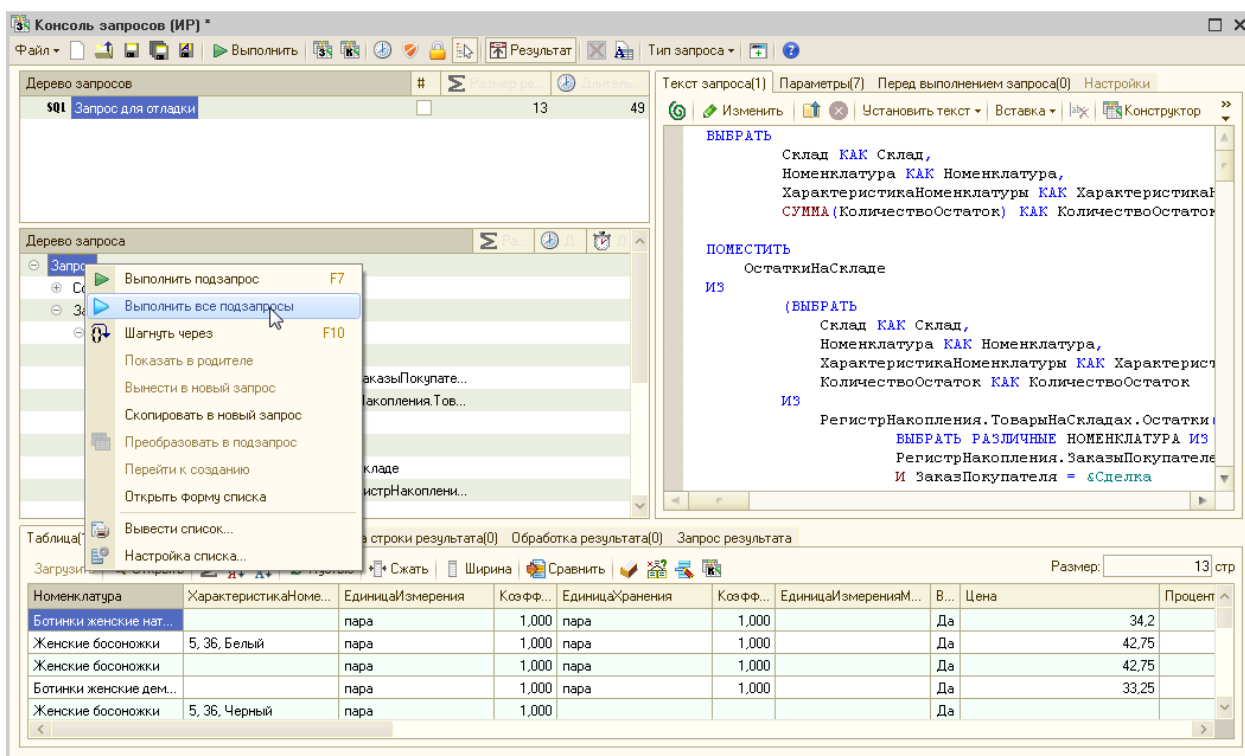
Вот этот запрос на создание временной таблицы. И дальше мы видим, что в дереве, вместо этого большого подзапроса, который мы сейчас смотрели - вставилось обращение к этой временной таблице.



Обращаю ваше внимание, что для того, чтобы все функции в работе с деревом запроса максимально правильно работали, нужно чтобы у всех временных таблиц были описаны псевдонимы. То есть, хотя платформа и поддерживает работу с временными таблицами без псевдонимов, но, к сожалению, анализ выполнять в дереве довольно сложно в этих случаях. Поэтому логика консоли опирается на то, что все псевдонимы у временных таблиц присутствуют. Если же они отсутствуют, то основная логика ломаться, не будет, но некоторые функции будут работать не полностью.

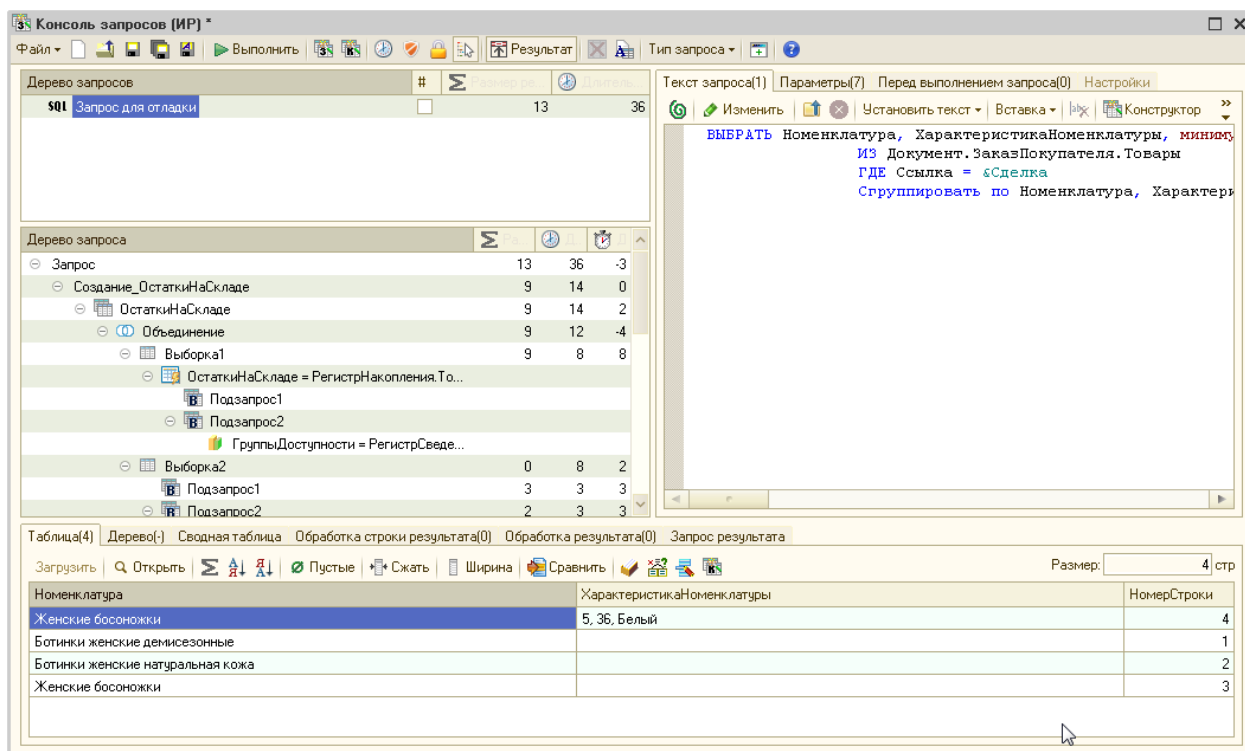
Вычисление критичных мест в запросе – команда «Выполнить все подзапросы»

Давайте теперь посмотрим на команду «Выполнить все подзапросы»



Я уже показывал, что если двойным щелчком кликнуть на любой веточке этого дерева, то выполняется соответствующий подзапрос.

И логично предположить, что вам когда-то захочется их все сразу выполнить, чтобы понять, где же здесь время чистое больше всего окажется.



Естественно, при вызове этой команды выполнились не все веточки, потому что некоторые ветки представляют собой обращение к таблицам непосредственно. Это значит, что если бы мы выполнили такой запрос, то вся таблица должна была бы выбираться. Ну а,

например, если у нас регистр бухгалтерии на 30 миллионов строк, то нет смысла выполнять такой запрос. Поэтому выполняются только подзапросы и выборки, участвующие в объединении, там, где чистая длительность может быть рассчитана.

Здесь конечно не очень показательно, потому что выбирается очень маленький объем данных (здесь даже где-то отрицательная чистая длительность получилась). Но, в общем, если запрос долгий, то вот эта колонка чистая длительность дополнительно сможет нам помочь в реальных ситуациях. То есть, здесь – на этом уровне могла бы быть большая чистая длительность, а все внутри запросы могли бы иметь длительность единицы. То есть, здесь допустим, какая-нибудь операция дополнительная накладывается, какой-нибудь отбор, из-за которого больше всего времени теряется в запросе. Но, повторюсь – это грубый инструмент поиска мест для оптимизации для тех, кому не хочется или нет возможности разбираться с планом запроса.

Вычисляемые параметры для запроса

Что тут еще можно посмотреть? В консоли запросов можно использовать вычисляемые параметры. Например, введем новый параметр ТекущийПользователь.

The screenshot shows the 'Консоль запросов (ИР) *' window. The 'Дерево запросов' pane on the left displays a query tree with a total duration of 13. The 'Параметры(8)' pane on the right lists various parameters. The 'ТекущийПользователь' parameter is selected, and its dropdown menu is open, showing 'Значение', 'Выражение', and 'Список'. The 'Выражение' option is highlighted.

Имя параметра	Тип	Значение	Тип ...	Не и...	Допустимые ти...
ГруппаДоступностиС...	Значение	Московские	Гру...	<input type="checkbox"/>	Группы доступн...
ДатаОстатков	Значение	04.04.2007 12:00...	Мом...	<input checked="" type="checkbox"/>	Дата
Договор	Значение	Договор Ин...	Дого...	<input type="checkbox"/>	Договоры конт...
Организация	Значение	Торговый до...	Орга...	<input type="checkbox"/>	Организации
Сделка	Значение	Заказ покуп...	Зака...	<input type="checkbox"/>	Заказ покупате...
Склад	Значение	Главный скл...	Скла...	<input type="checkbox"/>	Склады (места ...
СтатусТары	Значение	Возвратная ...	Стат...	<input type="checkbox"/>	Статусы партий...
ТекущийПользователь	Значение		Не о...	<input type="checkbox"/>	

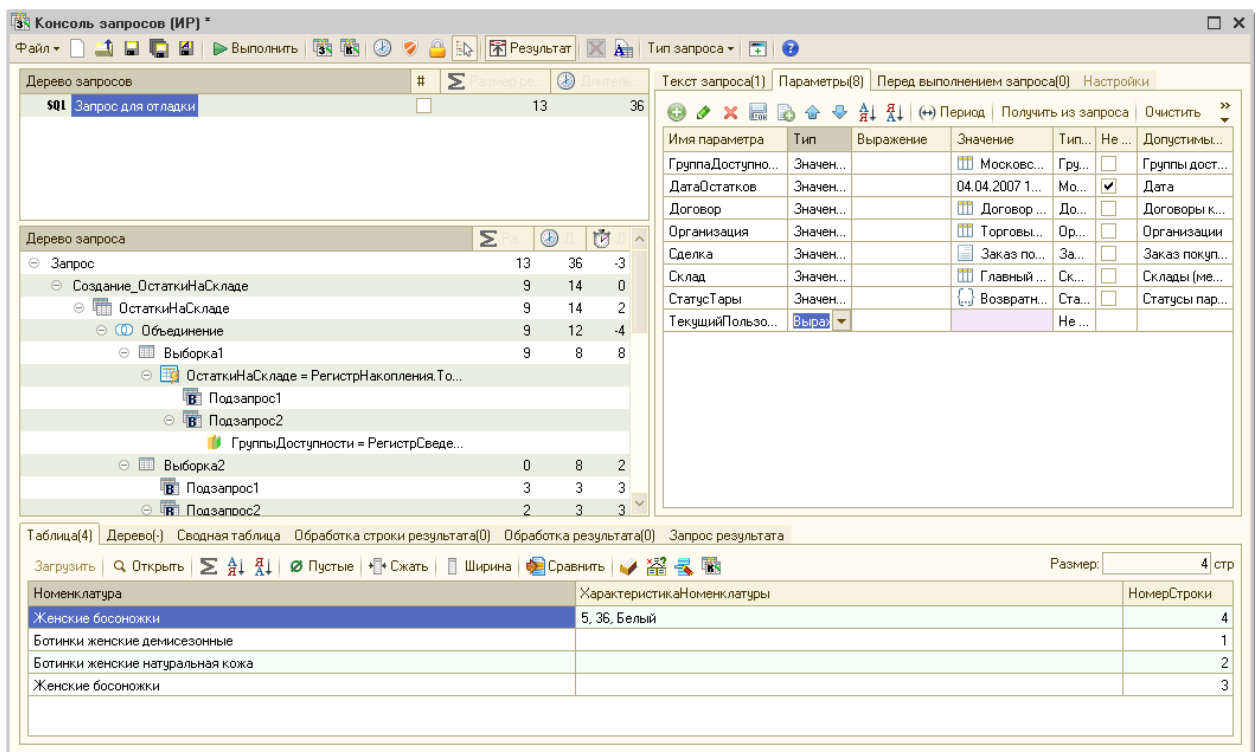
Dropdown menu for 'ТекущийПользователь':

- Значение
- Выражение
- Список

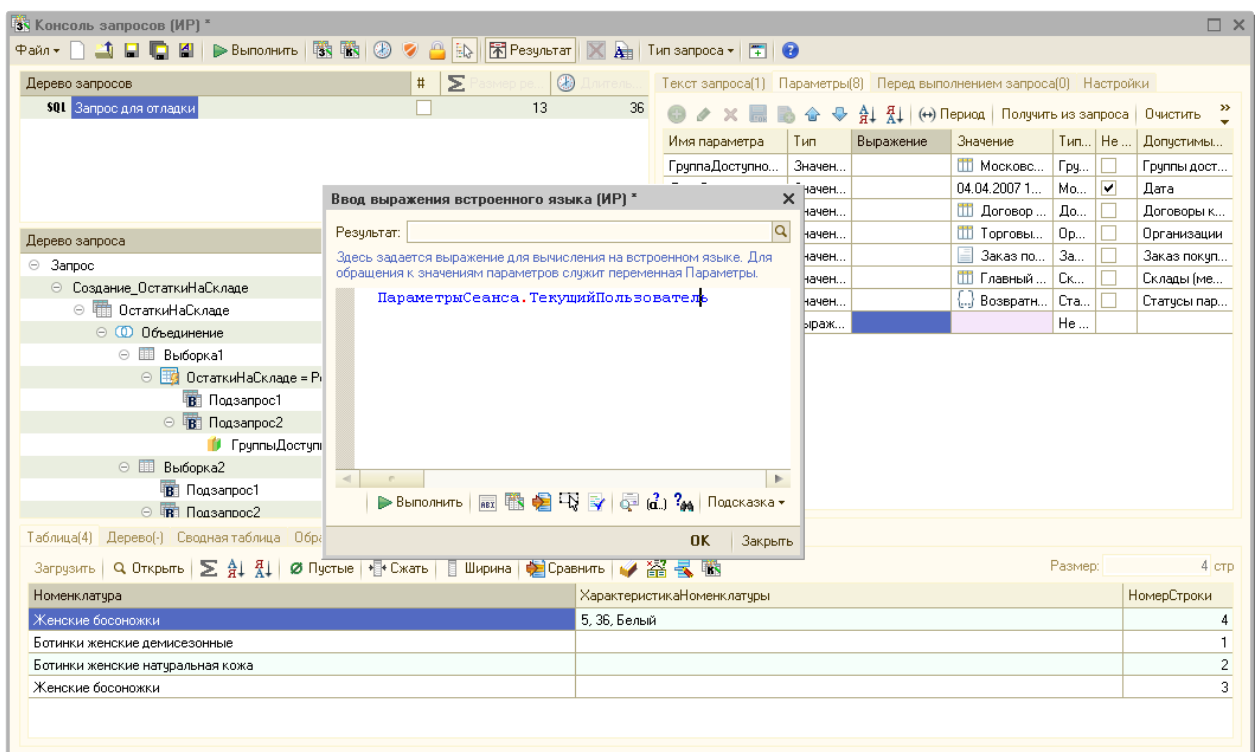
Table at the bottom:

Номенклатура	ХарактеристикаНоменклатуры	НомерСтроки
Женские босоножки	5, 36, Белый	4
Ботинки женские демисезонные		1
Ботинки женские натуральная кожа		2
Женские босоножки		3

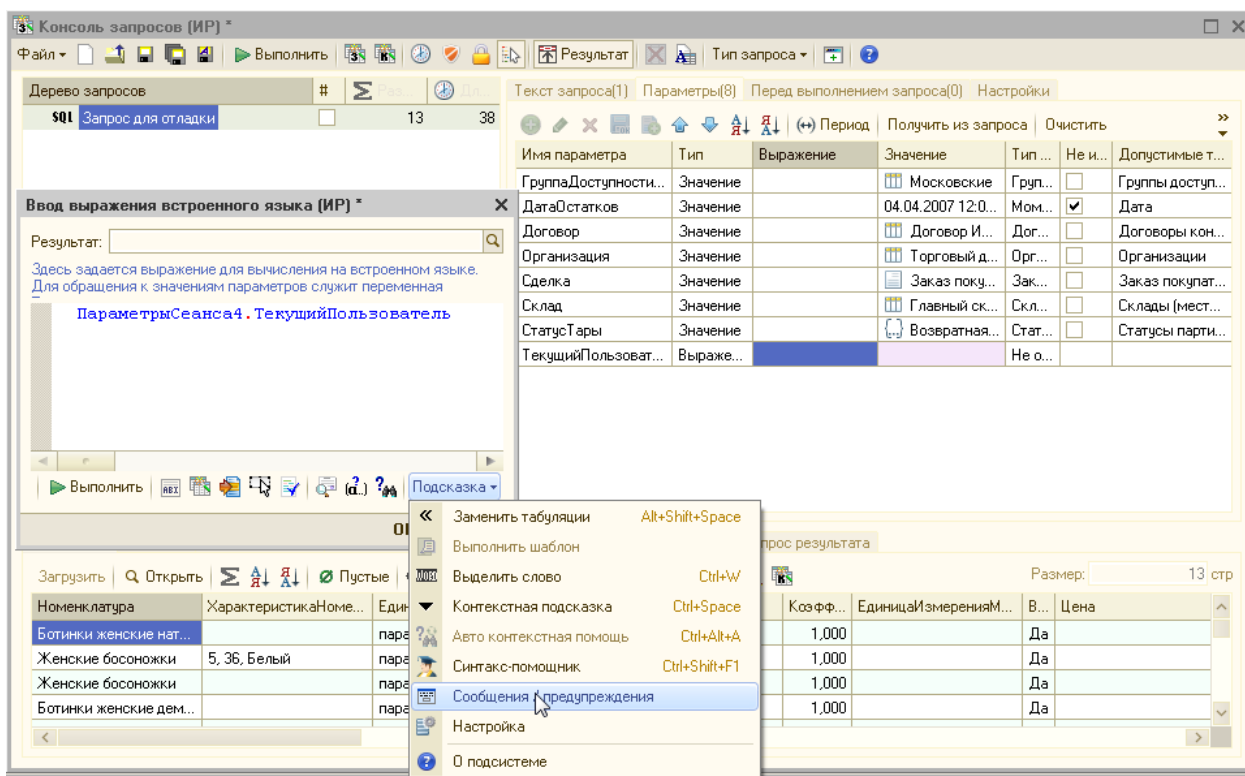
Тип параметра – выбираем «Выражение». Обратите внимание, у нас здесь появляется дополнительная колонка «Выражение» и ячейка значения для этого параметра подкрасилась фиолетовой, чтобы вы понимали, что значение не самостоятельное и его нельзя редактировать, т.к. оно вычисляется.



Откроем редактор выражения и напомним здесь –
 ПараметрыСеанса.ТекущийПользователь.



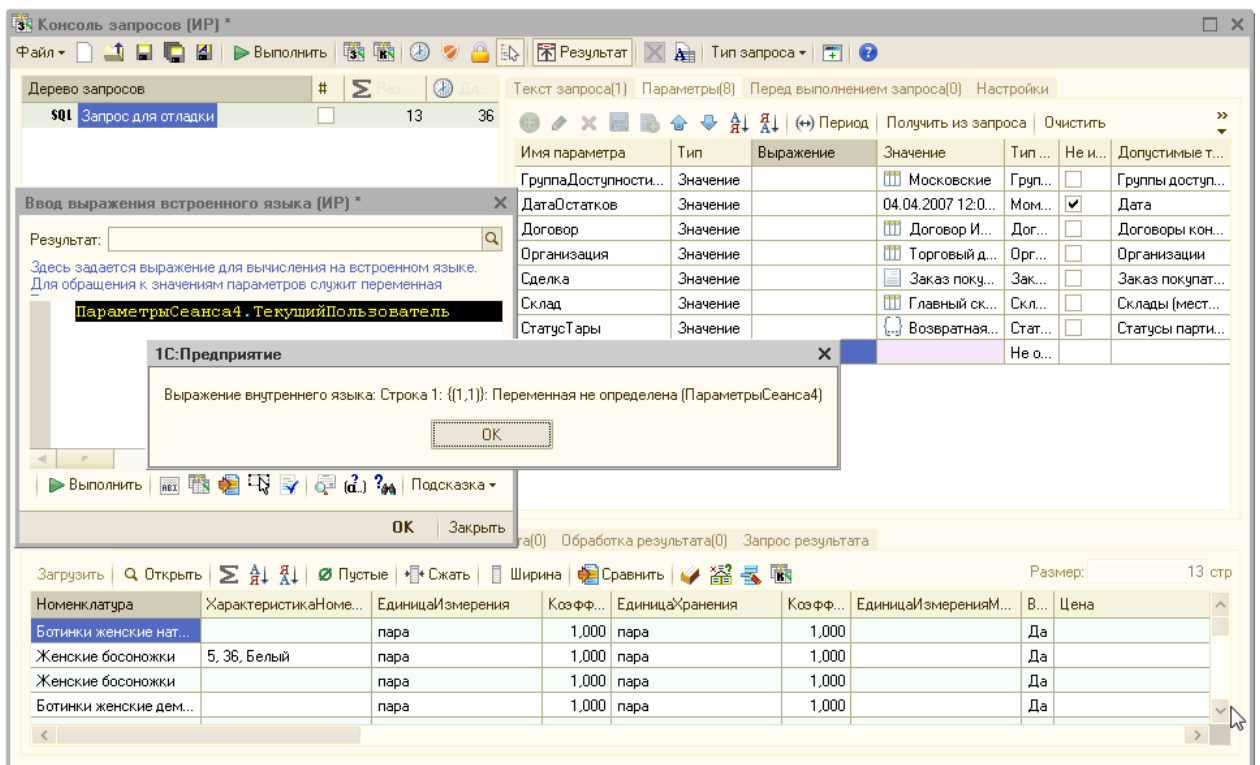
Мы написали выражение. Теперь, если нам хочется проверить, что мы все правильно написали, мы можем запустить синтаксический контроль. Например, если мы вместо ПараметрыСеанса напишем здесь что-нибудь вроде ПараметрыСеанса4, то он загружается.



Для того чтобы можно было здесь увидеть текст сообщений об ошибке, нужно включить режим Сообщения/предупреждения, чтобы сообщения выдавались модально.

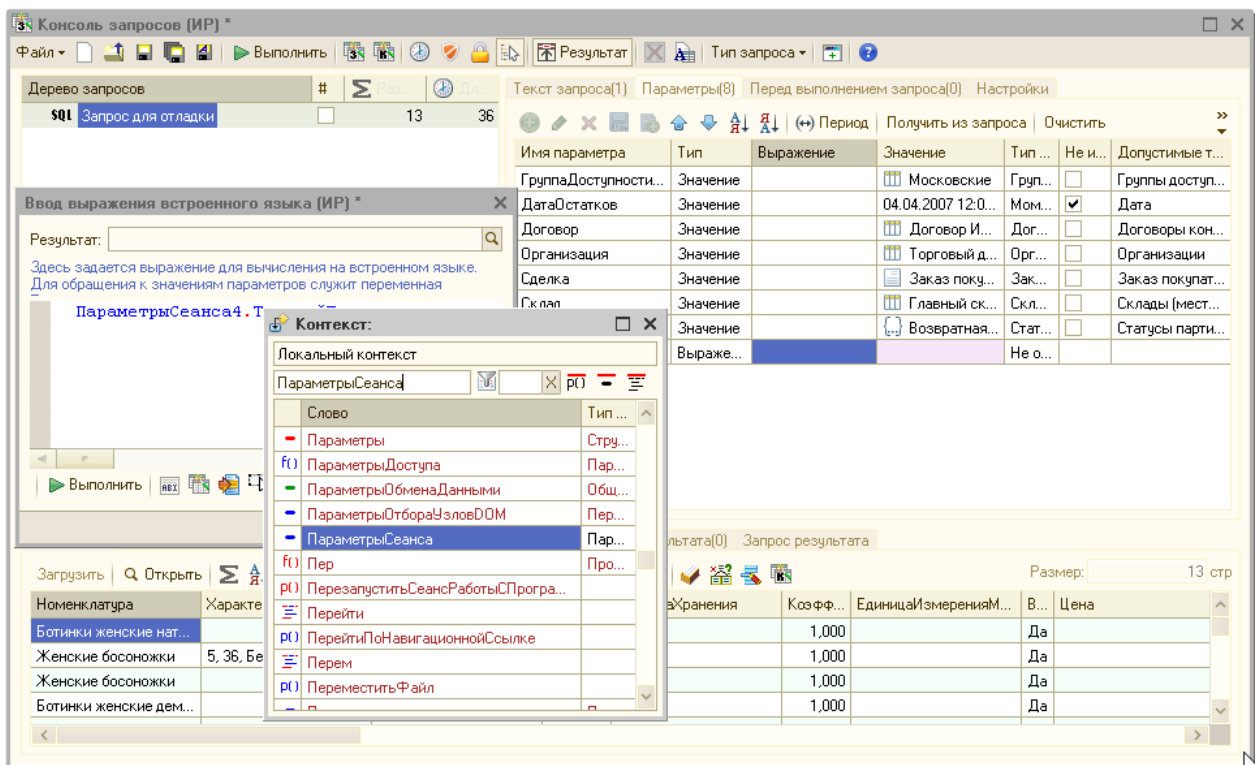
По умолчанию этот режим выключен, потому что, если по каждому сообщению, которое у нас выводится, у нас будет предупреждение выходить, то это скорее всего будет раздражать.

Давайте сейчас проверим синтаксис нашего выражения – нажмем кнопку «Синтаксический контроль» (Ctrl+F7).

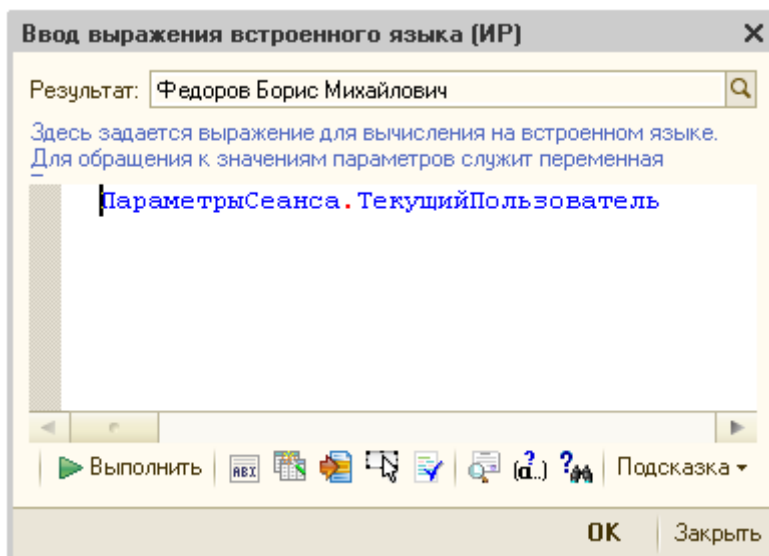


У нас выходит сообщение, что такой переменной нет.

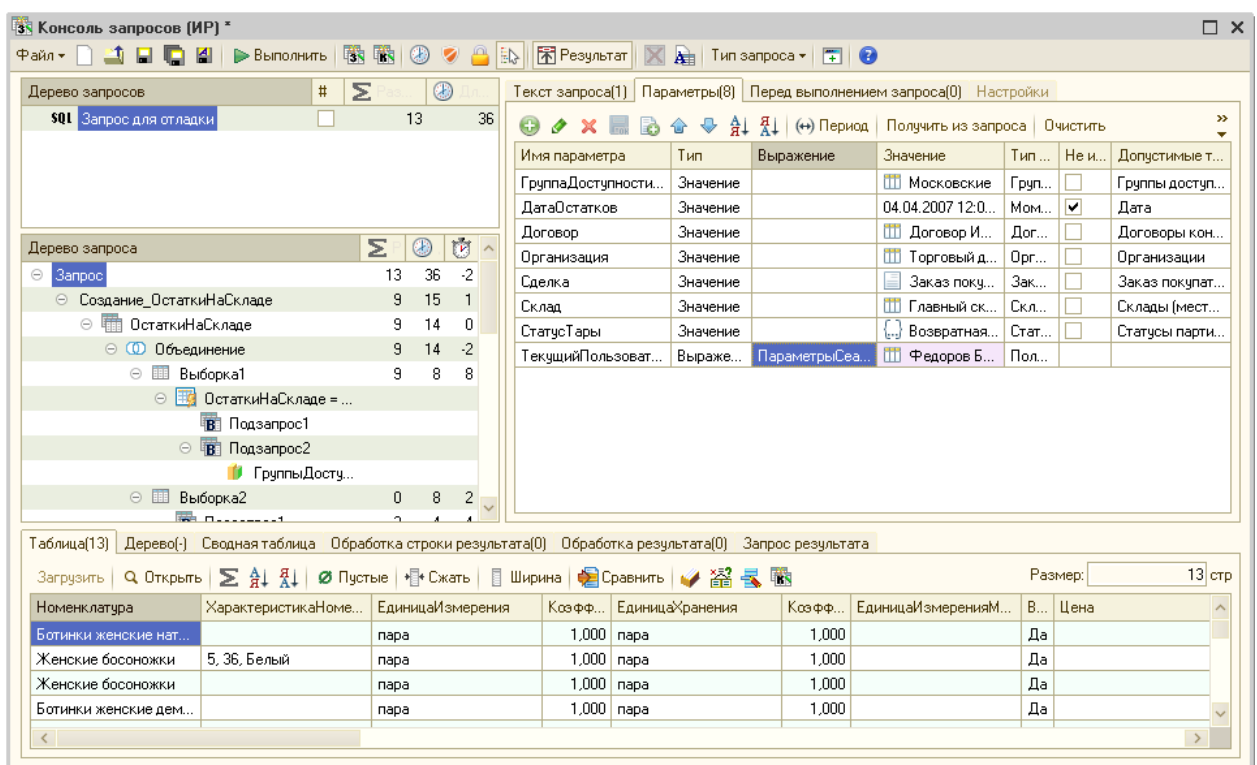
Я вызываю контекстную подсказку, смотрю, как правильно называется переменная и двойным щелчком исправляю выражение



Нажимаю кнопку «Выполнить» и вижу в поле «Результат» вычисленное значение.



Нажимаю ОК, и у меня в поле «Выражение» записывается то, что я сейчас редактировал, а в поле «Значение» попадает вычисленное значение.

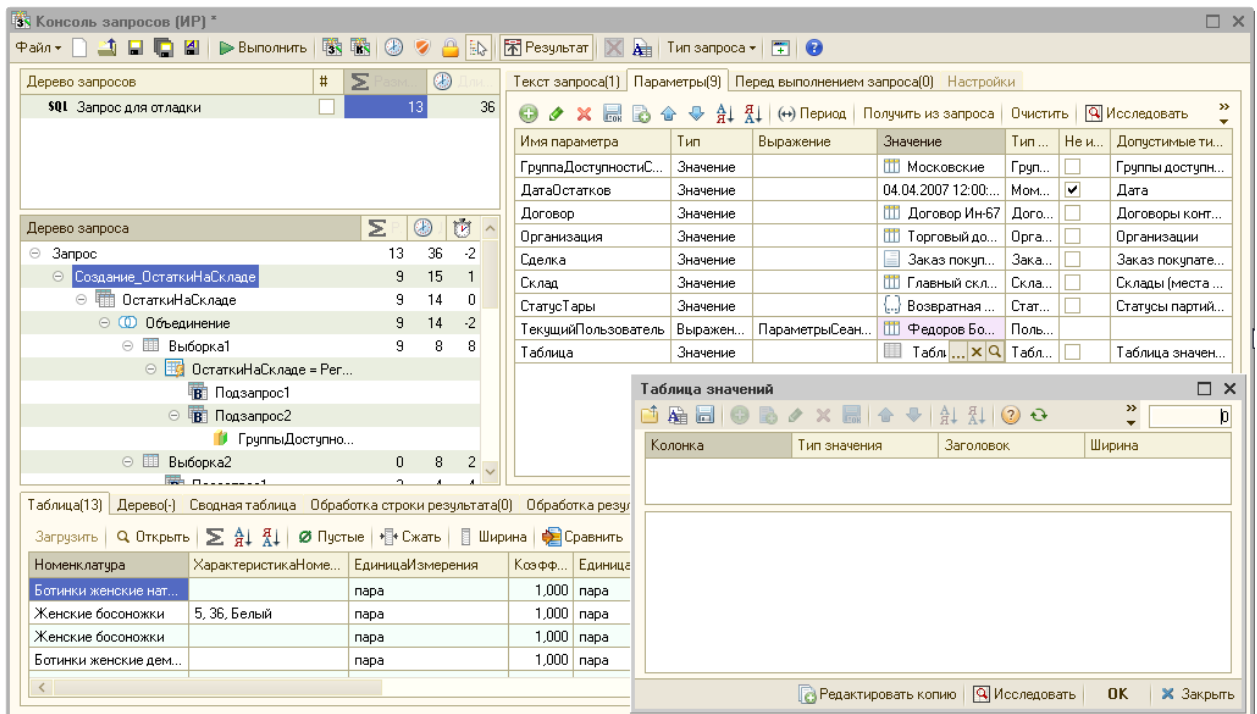


Двойным щелчком это значение не входит в режим редактирования, то есть, мы не сможем вручную изменить этот параметр.

Использование в качестве параметра запроса таблицы значений

Также в качестве параметра запроса вы можете задать таблицу значений, чтобы использовать ее внутри запроса. Например, для того, чтобы создать из нее временную таблицу. Либо можно проверять вхождение группы полей в эту таблицу с помощью задания условия Где несколько полей упорядоченных В – и дальше параметр и имя этой таблицы.

Для этого я создаю новый параметр – я его назвал «Таблица». Выбираю для него в качестве допустимого типа – тип ТаблицаЗначений. И после этого в поле «Значение» можно отредактировать саму таблицу.



Прерывание процесса выполнения кода через присвоение переменной некорректного значения (работа с функцией Пр).

Это очень простой пример, но он крайне полезный.

Допустим, у вас есть какой-то цикл долгий, в котором не вставлена ОбработкаПрерыванияПользователя. Вы его запустили, но вам захотелось его прервать. Обычно в таких случаях, придется либо долго ждать его завершения, либо завершить приложение 1С и потерять все настройки, которые вы не сохранили в нем.

Вот тут вам поможет функция Пр. Она выполняет присваивание значения переменной, что можно использовать для того, чтобы вызвать исключение прямо при выполнении текущей строки кода.

Что для этого нужно?

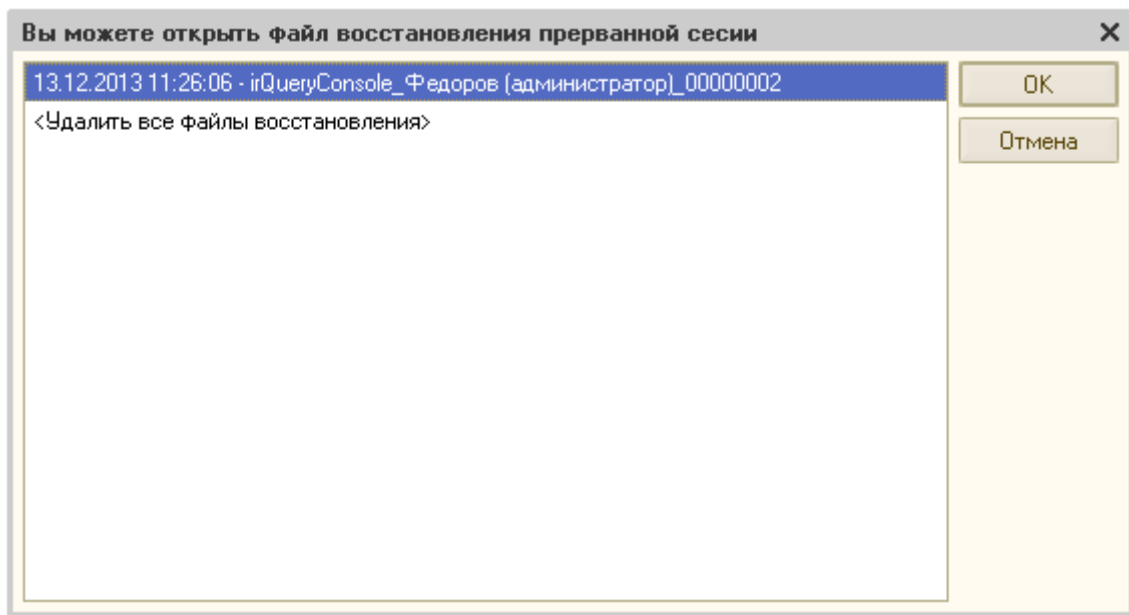
- Нажимаем Shift+F9, у нас текущая переменная подставилась сразу в ВычислитьВыражение.
- И пишем здесь: Пр(Переменная)
Здесь нам нужно присвоить нашей переменной какое-то такое значение, которое интерпретатор не сможет обработать и вызовет исключение. Второй параметр функции по умолчанию Неопределено, что вполне подходит для нашей цели.

Таким образом, я могу прервать выполнение программы и вернуть управление интерфейсу пользователя приложения. Хотя конечно же лучше все-таки там, где допускает логика, вставлять в коде вызов процедуры `ОбработкаПрерыванияПользователя`.

Демонстрация аварийного завершения консоли запросов

Банальная ситуация. Мы в консоли запросов написали какой-то запрос, не закрывали самой консоли, не сохраняли разработанный нами запрос в файл, и теперь, допустим, даже не аварийно завершил ее, а вот просто нажали в отладчике – перезапустить, например.

Что нам покажет консоль запросов при открытии?



Здесь у нас на выбор появилось два элемента списка.

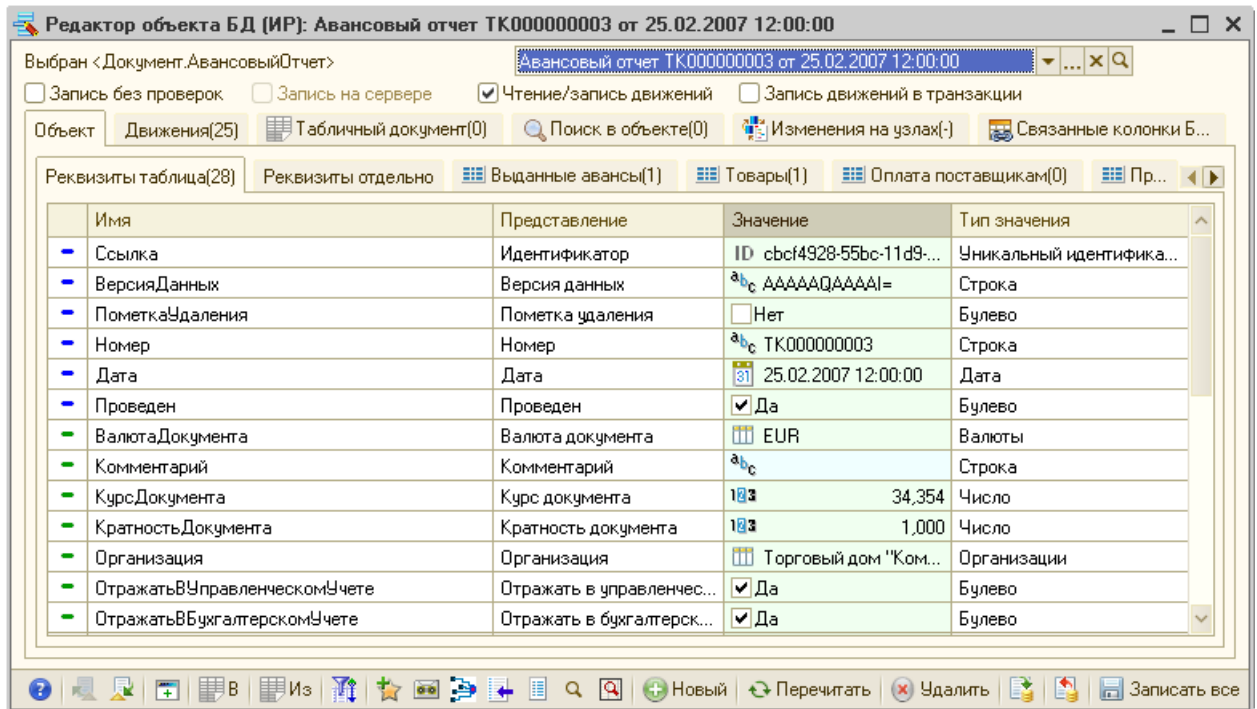
- Первое – это удалить все файлы восстановления – то есть, вам не нужно уже будет ничего ни сейчас, ни в будущем от того, что раньше было сохранено
- Второй вариант – выбрать этот файл, который был автоматически сохранен временный.
- И есть еще возможность отмену нажать – это третий вариант. Это значит, что сейчас вы не готовы сказать, что вы хотите сделать с файлом восстановления. Он будет сохранен на диске и в следующий раз, когда вы откроете консоль запросов – вам снова зададут этот же вопрос.

Восстановление после аварийного завершения поддерживается для тех инструментов подсистемы, которые наиболее подвержены риску аварийных завершений – это консоль кода и консоль запросов.

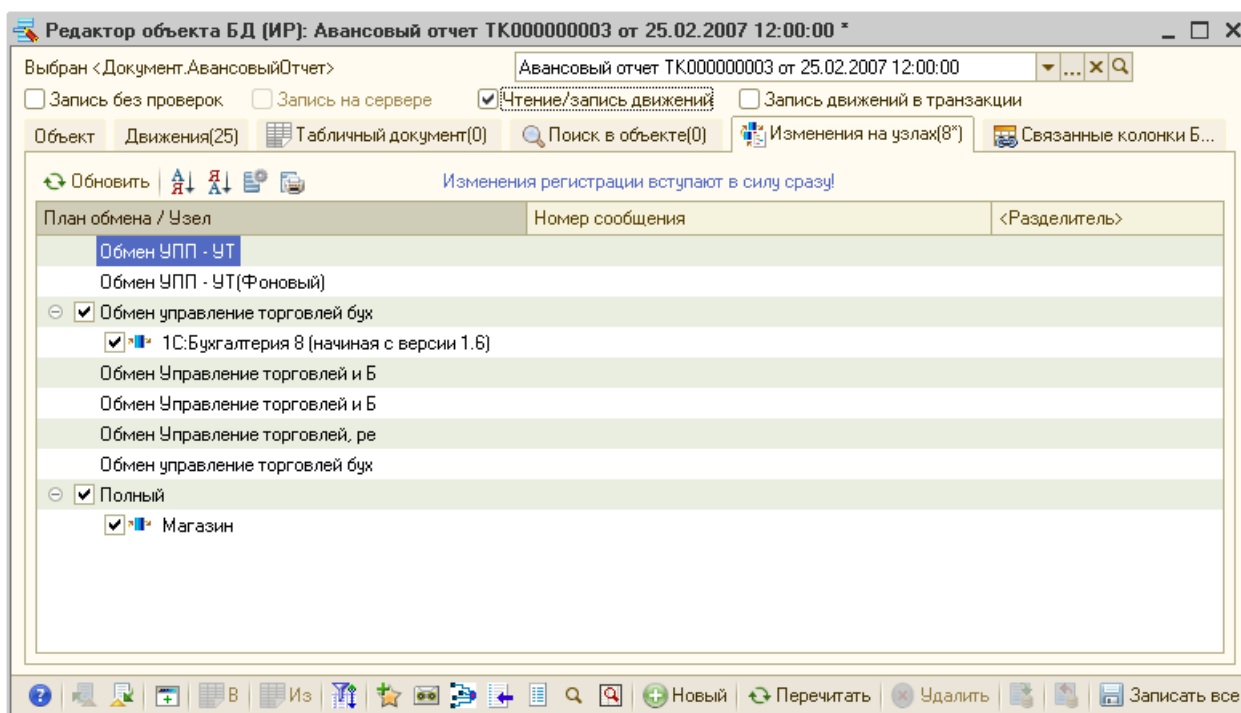
Редактор объекта

Давайте выберем еще одну какую-нибудь последнюю тему – либо работа с битыми ссылками, либо редактор объекта. Или разберем вариант «два в одном» - откроем редактор объекта и посмотрим, как в нем можно работать с битыми ссылками.

Вы можете убедиться, что редактор объекта гарантированно покажет вам все реквизиты, все табличные части, все движения, связанные с объектом.

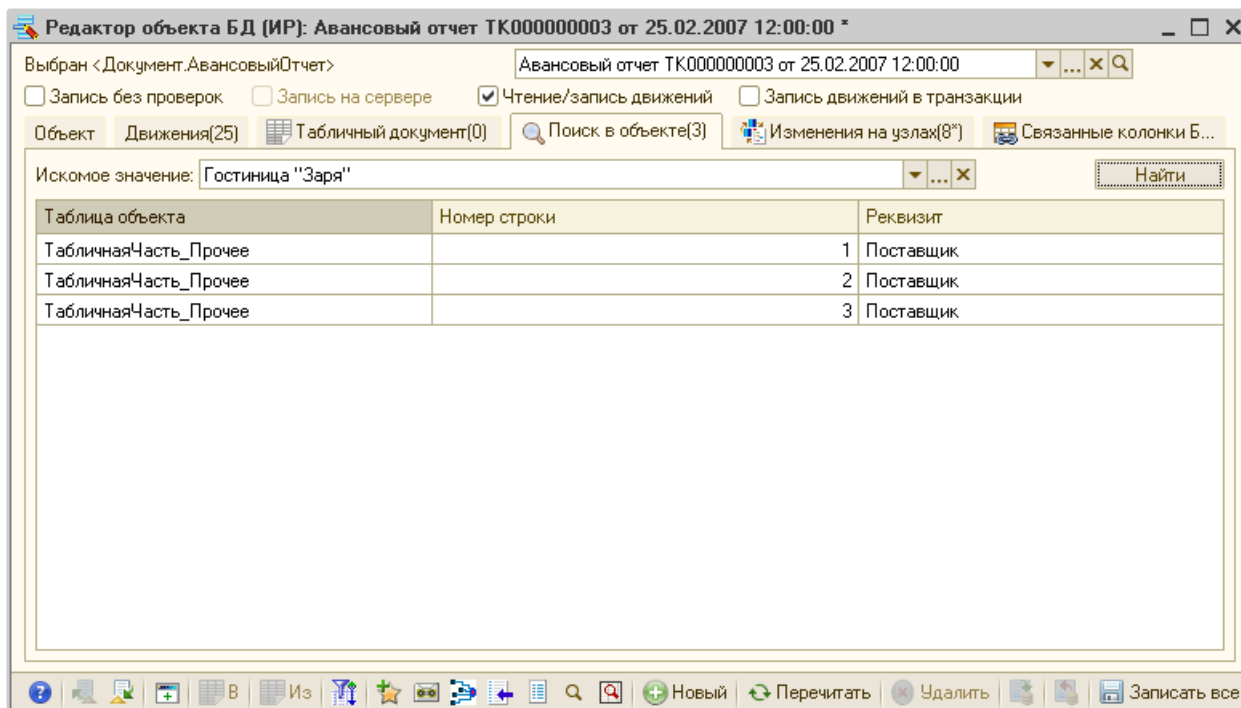


Кроме того, покажет даже регистрацию на узлах (записи в таблицах изменений). Это еще такая неявная составляющая, которая привязана к объекту.



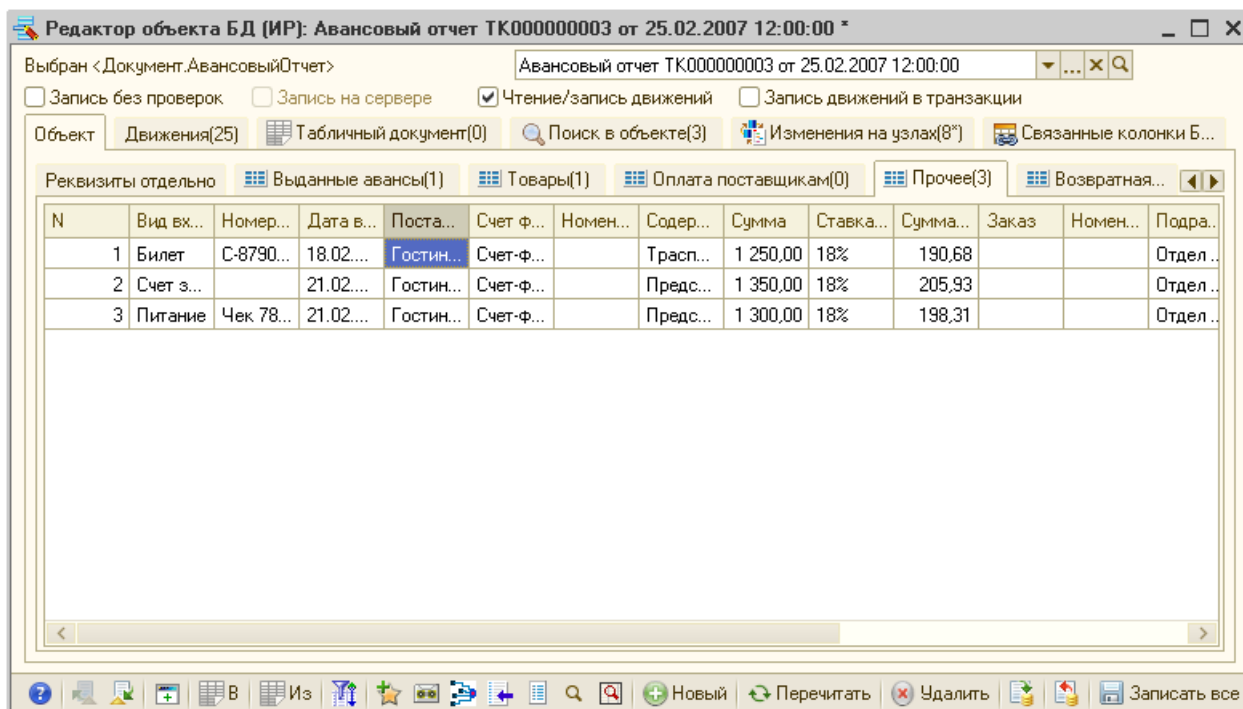
Какие еще возможности есть у этого редактора?

Например, вы открыли какой-то документ, у которого по регистру бухгалтерии 10 тысяч проводок – и вы хотите именно в движениях этого документа найти какую-то ссылку. Вы можете на закладке «Поиск в объекте» набрать код контрагента, и найти все его вхождения в этом объекте.

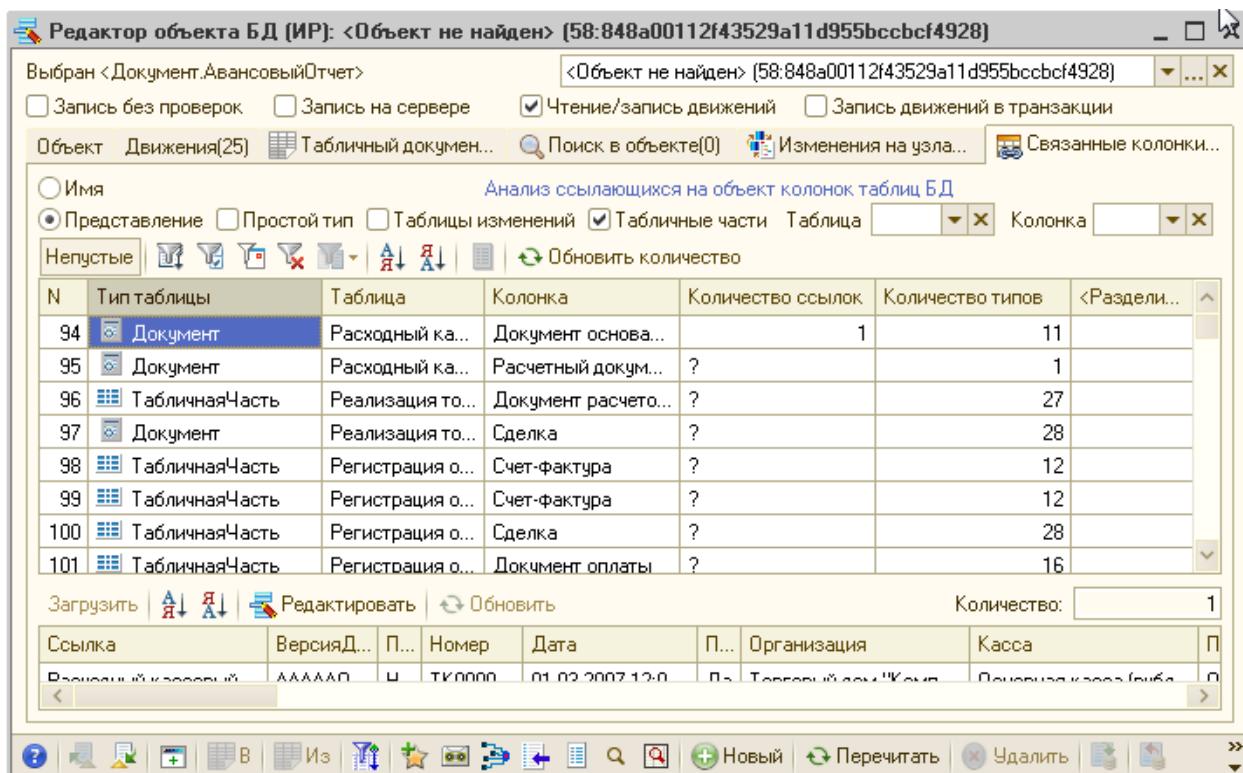


Выводите здесь это значение – и будут выбраны, показаны координаты всех вхождений этой ссылки в данный объект.

Причем, при выборе строки в таблице поиска в объекте вы сразу будете «перебрасываться» на выбранный номер строки конкретной табличной части в этом объекте, чтобы сразу увидеть всю связанную информацию.



Будет активизироваться именно нужная строка, нужная ячейка – прямо моментально все можно вручную там найти, изменить.



Есть еще интересная возможность – закладка «Связанные колонки базы данных». Здесь выполняется анализ, в каких колонках базы данных присутствует тип объекта. Тут также

можно будет посмотреть, какие строчки в этих таблицах ссылаются на объект. Это своего рода аналог инструмента «Поиск ссылок на объект», только более мощный.

А здесь вы можете нажать режим «Непустые», активизировать табличную часть, зажать кнопку «вниз» - и у вас по очереди для каждой связанной колонки будет проверяться наличие ссылок на этот объект.

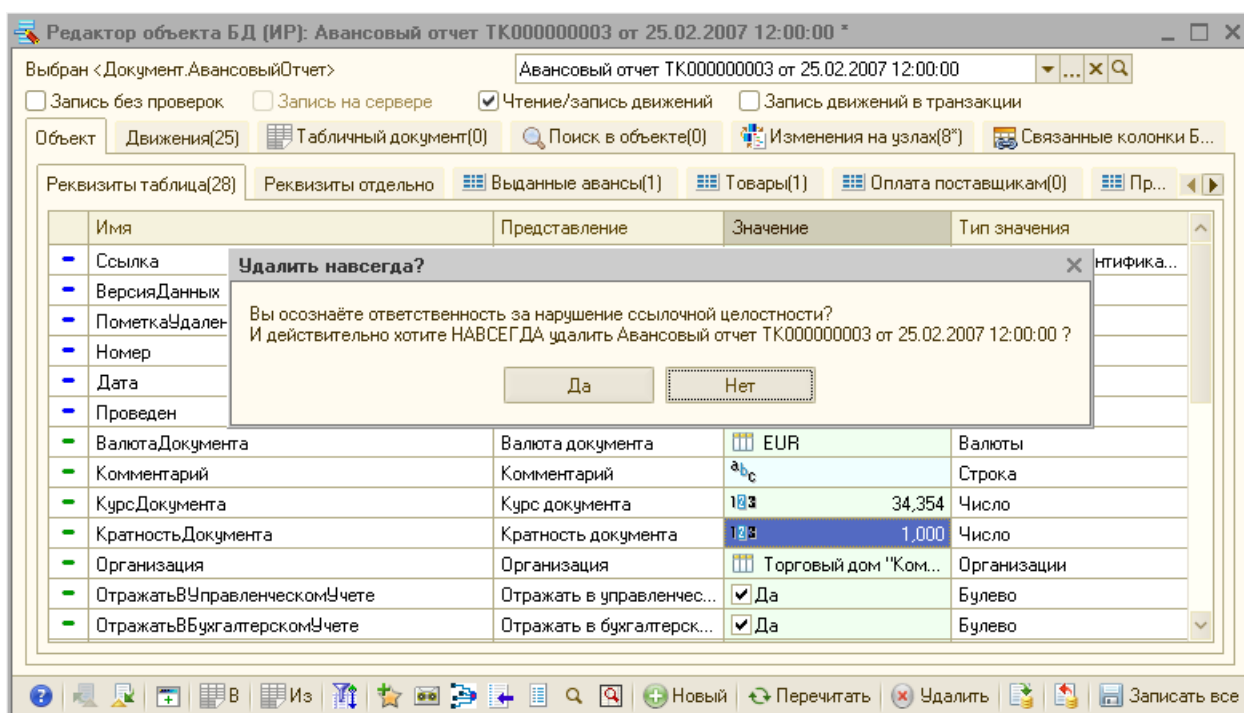
Все типы: 64%, Осталось: ~0:00:02.>> Анализ РегистрСведений: 1%, Осталось: ~

Обратите внимание, здесь снизу многоуровневый индикатор показан. Он тоже универсальный в подсистеме. Вы можете его сами тоже применять. Там удобно видеть время окончания и длительность. И везде, где есть такой индикатор, естественно, работает Ctrl+Break, чтобы вы могли прервать процесс.

Восстановление битых ссылок

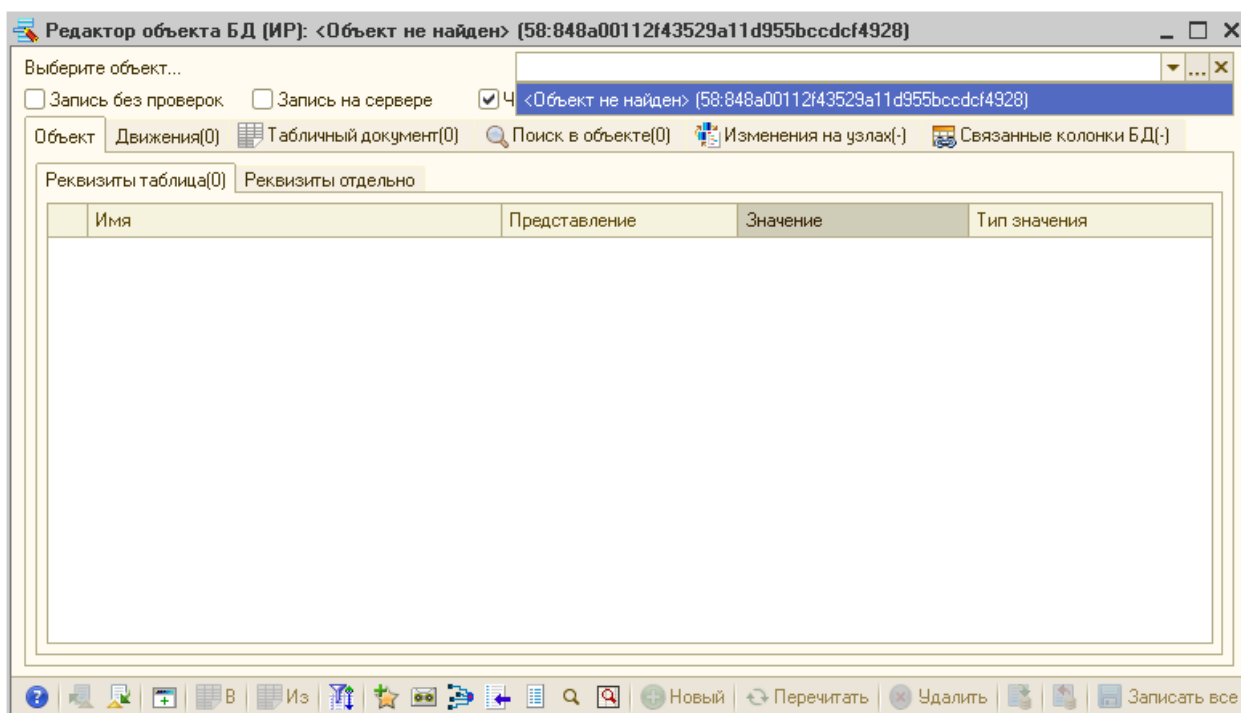
На самом деле для работы с битыми ссылками здесь есть поле Идентификатор. Оно называется «Ссылка», но, по сути – это идентификатор.

Я скопировал этот идентификатор в буфер обмена и сейчас – я удаляю этот документ.

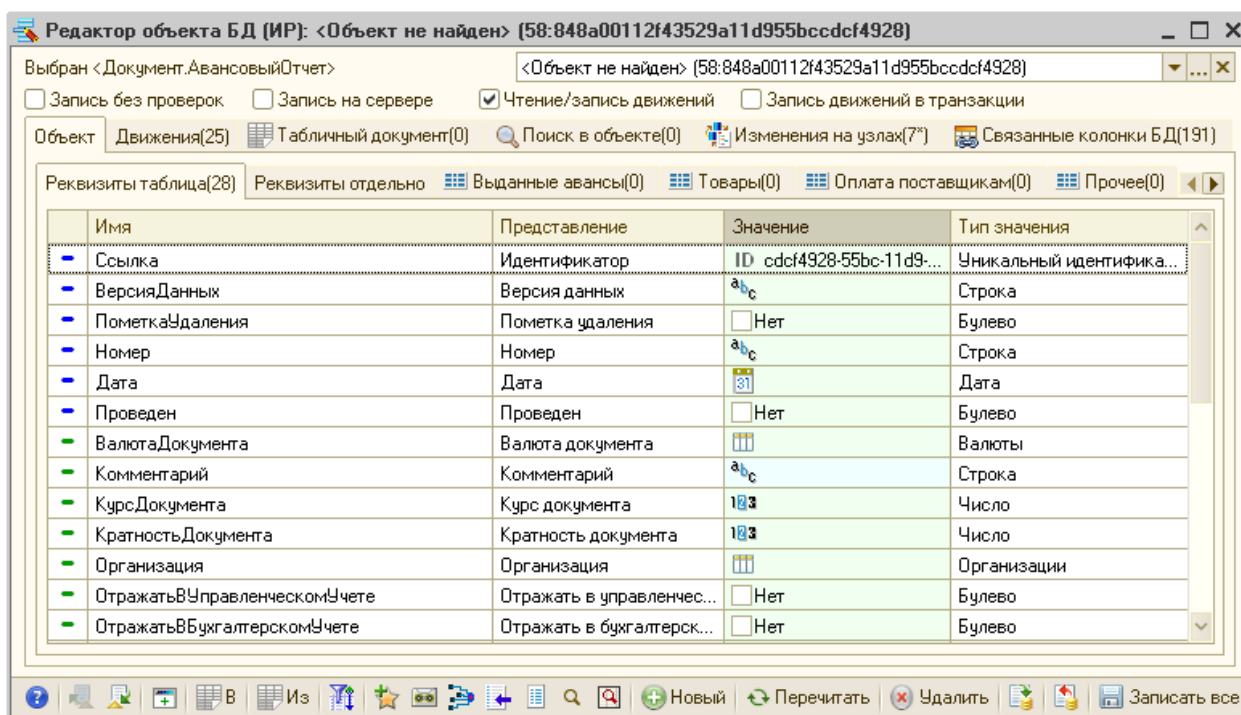


Он удалился.

Теперь – я выбираю тип.



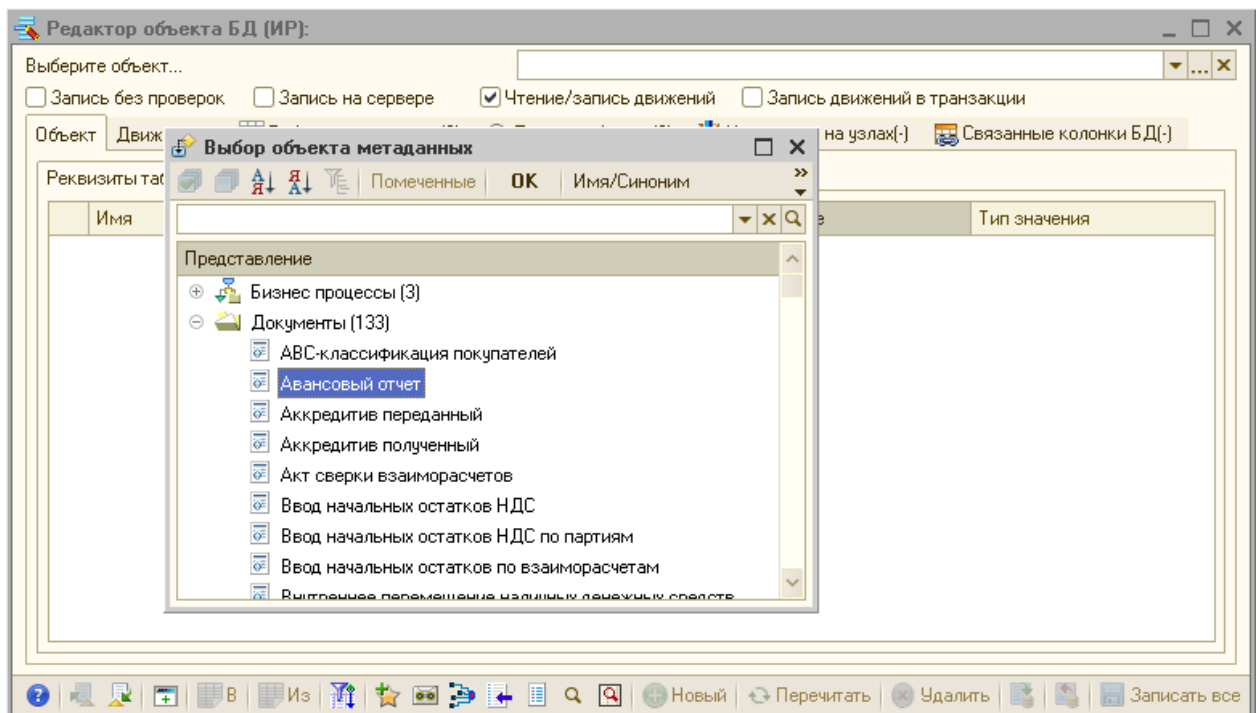
В списке последних использованных объектов он показан как <Объект не найден> (58:848a00112f43529a11d955bccbcf4928).



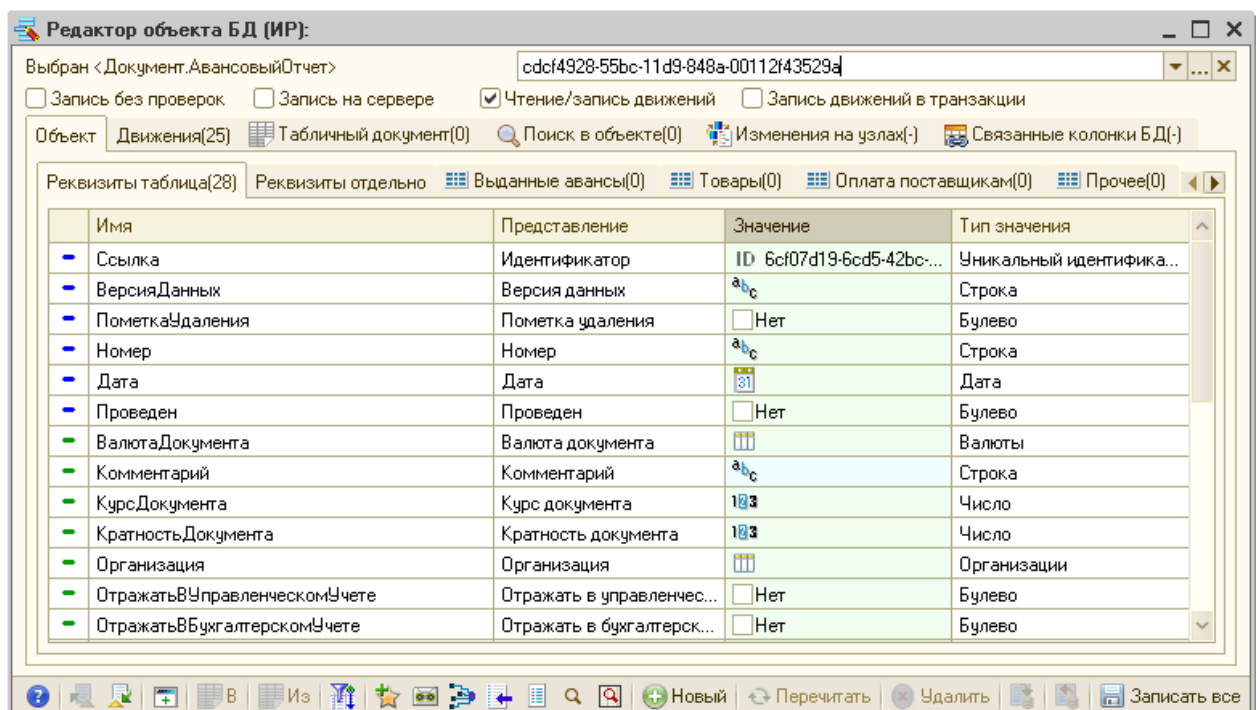
Давайте его выберем. Как видите, все появилось. Для него даже будут считаны движения. То есть, вы сможете работать с движениями документа, даже если он отсутствует. Однако все реквизиты документа естественно имеют пустые значения.

Посмотрите внимательно на эту ссылку <Объект не найден> (58:848a00112f43529a11d955bccbcf4928). 58 – это внутренний номер таблицы, то есть это

число описывает тип объекта. И после двоеточия уже идет идентификатор конкретной записи в его таблице.



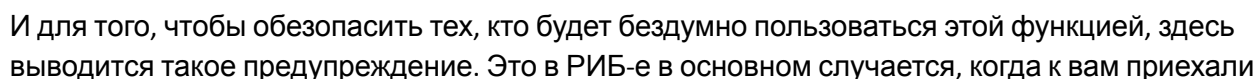
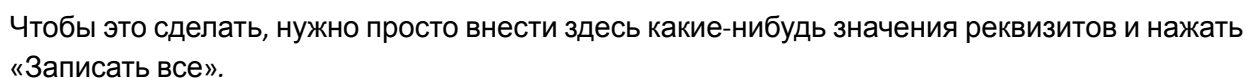
Я очищаю это поле – видите, здесь сейчас не выбран тип. И теперь я могу через кнопку выбора вида метаданных найти там таблицу восстанавливаемого объекта и потом вставить в поле выбора объекта его идентификатор.



Откроется именно тот самый «потерянный» объект.

А в случае, если я вставляю строку <Объект не найден> (58:848a00112f43529a11d955bccbcf4928) именно в таком виде, то даже тип сразу

Так как же восстановить из битой ссылки документ и его движения?



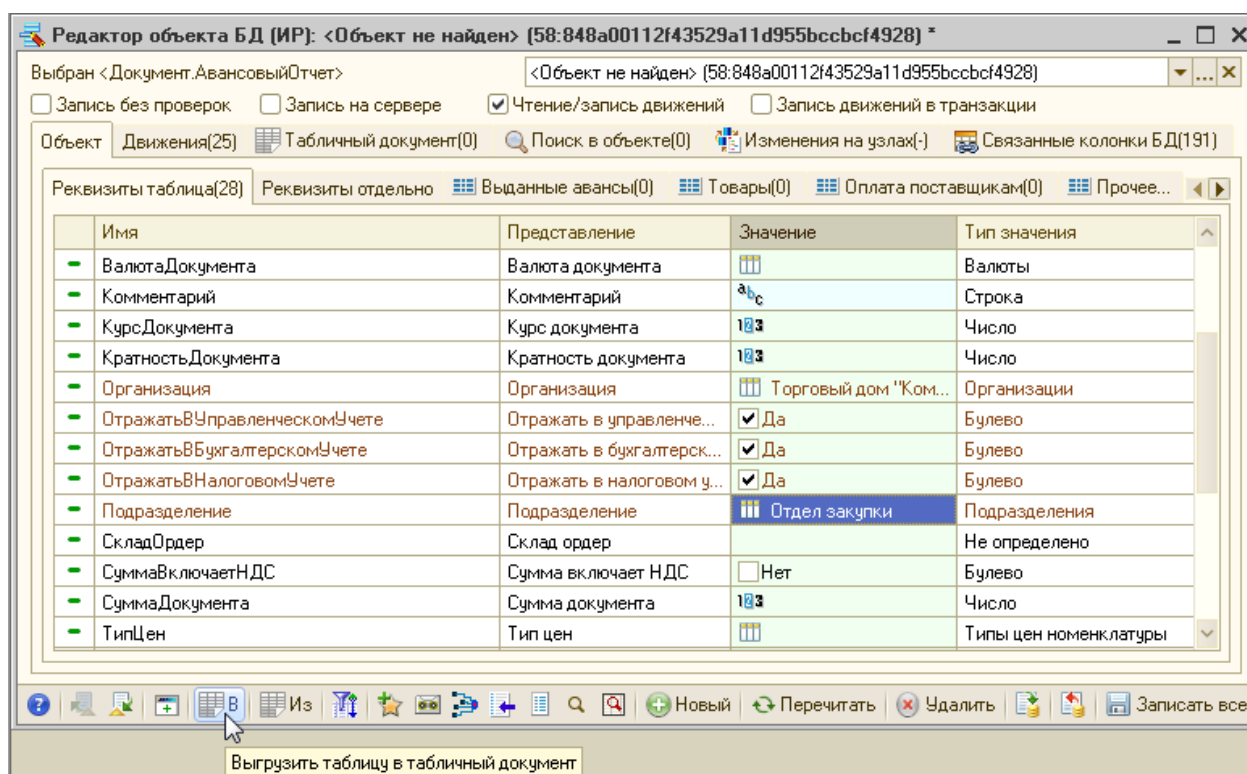
какие-то движения от документа, который еще отсутствует в вашей базе. И если вы здесь его раньше времени создадите, то потом, когда он по РИБу придет из другой базы – у вас возникнет конфликт и неизвестно, что в итоге останется. Вы можете сами себе навредить.

Но если вы полностью уверены в том, что вы делаете, то продолжаем.

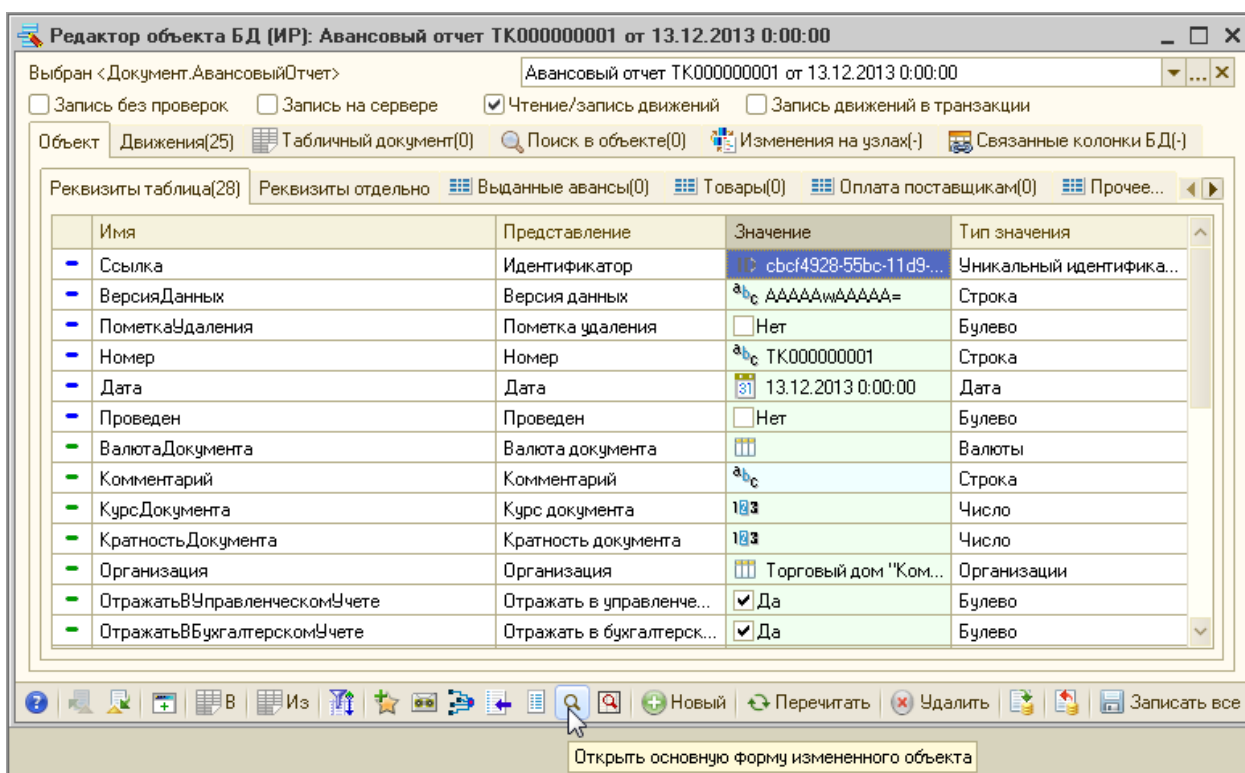
- **А в этом редакторе есть сериализация, чтобы нажать кнопку, скопировать текст и потом его в другом месте вставить?**

— Непосредственно такой возможности действительно нету, хотя кто-то уже писал ней мне такое пожелание. Есть возможность вывести все данные в табличный документ (или загрузить их из табличного документа). Но в этот табличный документ, к сожалению, помещаются не все данные документа, а только данные текущей таблицы. То есть, если вы в таблице реквизитов находитесь, то туда будет помещена таблица реквизитов, если в таблице табличной части, то табличная часть. Если в каком-то движении, то она вам туда движения загрузит. Но вот все вместе – такого, к сожалению, нет. Но пожелание хорошее и я его обязательно реализую.

28.06.2014: Позже сериализация и десериализация данных объекта была реализована.



Итак, сейчас мы с вами восстановили документ. Хотя не совсем, конечно, восстановили, потому что просто заполнили навскидку какими-то случайными значениями. Но главное, что он появился, и вы теперь сможете зайти в его форму.



Нажимаете кнопку «Открыть основную форму измененного объекта» и можете работать с документом дальше уже в родном интерфейсе.

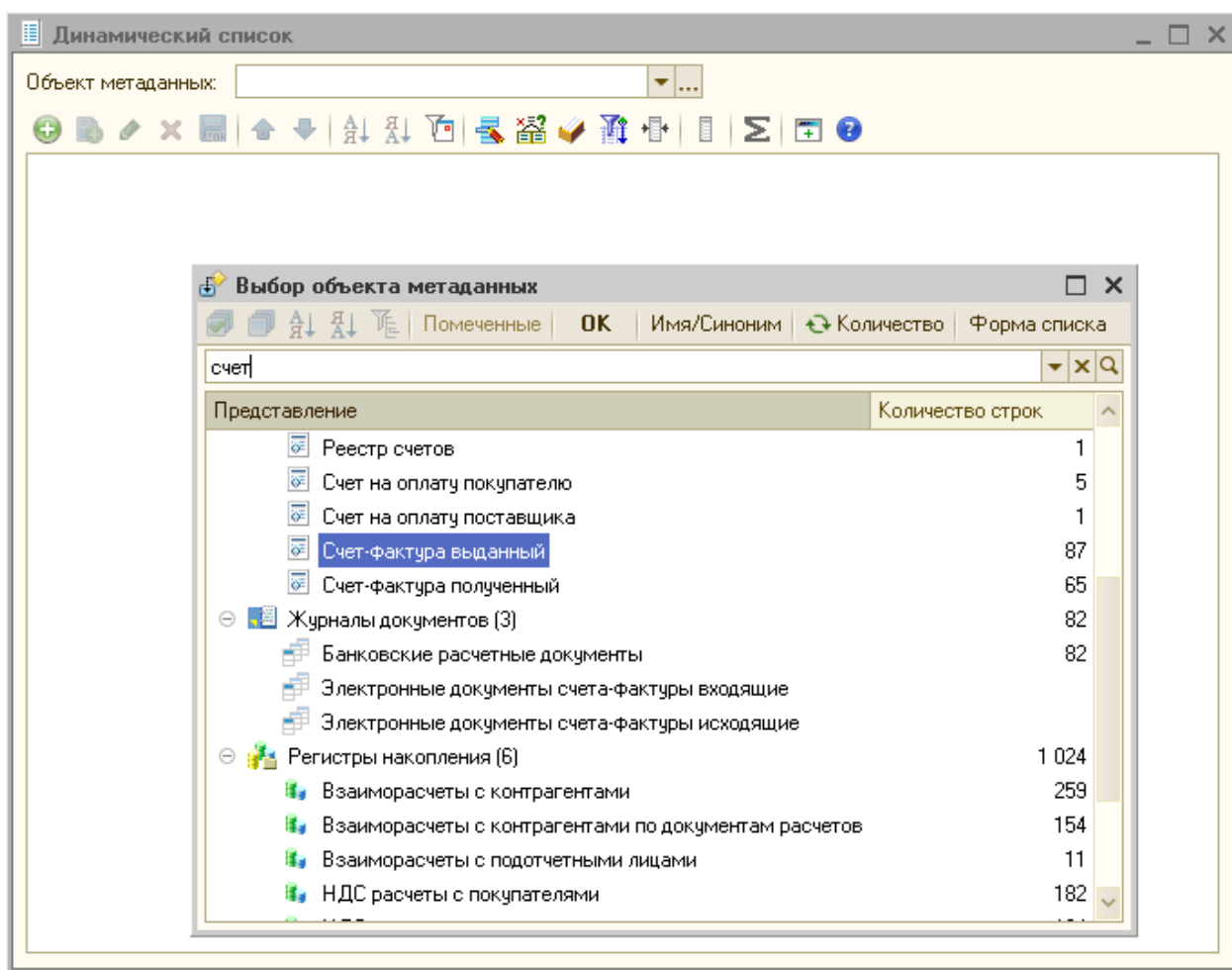
Динамический список.

В обычных формах списка у вас часть метаданных закрыта какими-то правилами, отбор может быть включен какой-то недоступный, часть реквизитов не показана, отсутствует возможность прямого доступа. То есть, интерфейсы загораживают нам, разработчикам, работу с данными.

Поэтому в подсистеме «Инструменты разработчика» реализован специальный инструмент – динамический список. С его помощью вы можете для выбранного объекта метаданных в табличном поле просмотреть все данные элементов этого объекта без всяких ограничений.

Этот инструмент реализован именно для разработчика, чтобы дать ему полный доступ ко всем данным.

Сначала здесь надо выбрать объект метаданных через поле ввода.



Кстати, здесь есть хорошая возможность для обзора объема таблиц базы. При нажатии на кнопку «Количество» будет показано количество строк в таблицах, но конечно придется немного подождать.

Расширенный отбор в динамическом списке

Здесь есть расширенный отбор, альтернативный стандартному, который модально открывается. Модальность стандартного диалога меня всегда очень возмущала, потому что при одновременной доступности обеих форм открывается много удобств.

Табличное поле: ДинамическийСписок		Документ список: Заказ покупателя	
Тип связанного табличного поля		Объект метаданных: Документ.ЗаказПокупателя	
Документ список: Заказ покупателя (двойной клик. обн...)		Действия	
Отбор - связанный	Порядок: применяемый		
Выключить все	Режимы	Инертировать	
Без отбора.			
Поле	Тип сравнения		
Значение	П.		
С	По		
<input type="checkbox"/> Документы по договору к...	Равно		
<input type="checkbox"/> Документы по контрагенту	Равно		
<input type="checkbox"/> Связанные документы	Равно		
<input type="checkbox"/> Номер	Содержит		
<input type="checkbox"/> Дата	Интервал (>=, <=)		
<input type="checkbox"/> Ссылка	Равно		
<input type="checkbox"/> Понетка удаления	Равно		
<input type="checkbox"/> Проведен	Равно		
<input type="checkbox"/> Адрес доставки	Содержит		
<input type="checkbox"/> Валюта документа	Равно		

Номер	Дата	Провер...	Понетка у...	Адрес дост...	Валюта до...	Удалить вр...	Дата оплаты	Дата отгру...	Дисконтна...	Договор к...	Использов...	Итоговая п...	Кол...
TK0000...	01.02.2007 21:10...	✓	Нет		USD		01.02.2007	06.02.2007		Договор к...	Да	912,35	
TK0000...	03.02.2007 11:42...	✓	Нет		руб.		08.02.2007	07.02.2007		Договор к...	Да	1 125,72	
TK0000...	03.02.2007 14:00...	✓	Нет		USD		03.02.2007	06.02.2007		Договор к...	Да	912,35	
TK0000...	10.02.2007 12:00...	✓	Нет		USD		13.02.2007	15.02.2007		Договор к...	Нет	4 149,76	
TK0000...	11.02.2007 0:00:00	✓	Нет		USD		16.02.2007	18.02.2007		Договор И...	Да	1 303,68	Отп...
TK0000...	13.02.2007 12:00...	✓	Нет		USD		18.02.2007	20.02.2007		Договор к...	Да	2 156,05	
TK0000...	13.02.2007 21:18...	✓	Нет		USD		18.02.2007	15.02.2007		Договор п...	Да	4 149,76	
TK0000...	14.02.2007 21:18...	✓	Нет		EUR		19.02.2007	20.02.2007		Договор к...	Да		
TK0000...	15.02.2007 12:00...	✓	Нет		руб.		20.02.2007	21.02.2007		Договор к...	Да	40 083,56	
TK0000...	17.02.2007 12:00...	✓	Нет		руб.		22.02.2007	23.02.2007		Договор к...	Да	17 889,16	
TK0000...	20.02.2007 12:00...	✓	Нет		руб.		22.02.2007	22.02.2007	Дисконтна...	Договор к...	Да	98 398,44	
TK0000...	01.03.2007 12:00...	✓	Нет		руб.		11.03.2007	07.03.2007		Договор к...	Да		
TK0000...	04.03.2007 12:00...	✓	Нет		EUR		09.03.2007	10.03.2007		Договор к...	Да	1 857,42	
TK0000...	04.03.2007 12:00...	✓	Нет		USD		09.03.2007	10.03.2007		Договор к...	Нет		
TK0000...	05.03.2007 12:00...	✓	Нет		USD		10.03.2007	11.03.2007		Договор к...	Нет		
TK0000...	06.03.2007 12:00...	✓	Нет		USD		11.03.2007	12.03.2007		Договор к...	Да		
TK0000...	07.03.2007 12:00...	✓	Нет		USD		12.03.2007	13.03.2007		Договор к...	Да	2 389,68	
TK0000...	08.03.2007 12:00...	✓	Нет		USD		13.03.2007	13.03.2007		Договор к...	Да	2 389,68	
TK0000...	11.03.2007 12:00...	✓	Нет		USD		16.03.2007	17.03.2007		Договор к...	Да		
TK0000...	12.03.2007 0:00:00	✓	Нет		USD		15.03.2007	17.03.2007		Договор к...	Да		
TK0000...	12.03.2007 12:00...	✓	Нет		USD		17.03.2007	18.03.2007		Договор к...	Да		
TK0000...	13.03.2007 12:00...	✓	Нет		USD		18.03.2007	17.03.2007		Договор к...	Да		
TK0000...	17.03.2007 12:00...	✓	Нет		EUR		22.03.2007	23.03.2007		Договор к...	Да		
TK0000...	01.04.2007 13:18...	✓	Нет		руб.		05.04.2007	07.04.2007		Договор к...	Да		
TK0000...	02.04.2007 12:00...	✓	Нет		USD		16.05.2007	04.04.2007		Договор И...	Нет		

Здесь элементы отбора представлены в виде таблицы, и, например, можно через Ctrl+F найти нужный элемент по подстроке. Согласитесь, это гораздо удобнее стандартного диалога, где поиск не предусмотрен и если элементов отбора очень много, то приходится долго скролить и искать нужный, читая название каждого. Также при вызове этого диалога в таблице отбора текущей строкой устанавливается элемент текущей колонки списка.

Здесь можно сразу нажать кнопочку « << » и получить отбор из текущей строки, чтобы установить отбор по текущему значению (хотя, конечно это дублирование стандартного функционала – отбора по текущей колонке).

Настройка сортировки динамического списка

Здесь вы можете задать порядок, причем использовать даже те поля, которые недоступны в штатном диалоге.

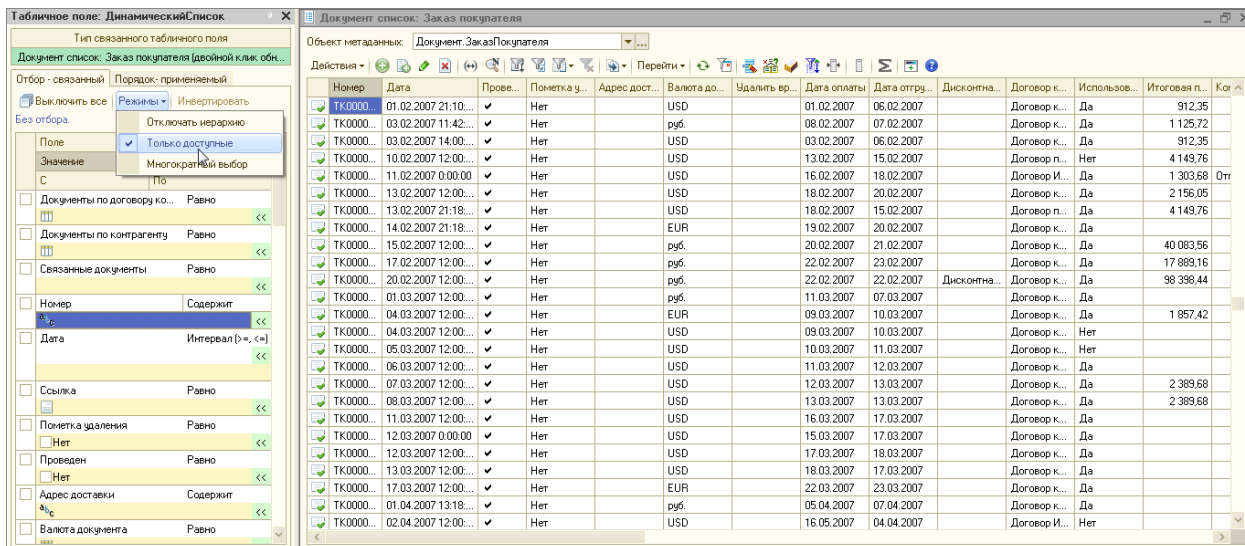
Табличное поле: ДинамическийСписок		Документ список: Заказ покупателя	
Тип связанного табличного поля		Объект метаданных: Документ.ЗаказПокупателя	
Документ список: Заказ покупателя (двойной клик. обн...)		Действия	
Отбор - связанный	Порядок: применяемый		
Применить	Очистить		
Поле	Направление сортиров...		
✓ - Дата	По возрастанию		
Доступные поля			
<ul style="list-style-type: none"> Итог плановая себестоимость Комментарий Контактное лицо контрагента Контрагент Кратность взаиморасчетов Курс взаиморасчетов Номер Номер входящего документа с электронного о... Организация Ответственный Подразделение Понетка удаления Проведен Склад группа Ссылка 			

Номер	Дата	Провер...	Понетка у...	Адрес дост...	Валюта до...	Удалить вр...	Дата оплаты	Дата отгру...	Дисконтна...	Договор к...	Использов...	Итоговая п...	Кол...
TK0000...	01.02.2007 21:10...	✓	Нет		USD		01.02.2007	06.02.2007		Договор к...	Да	912,35	
TK0000...	03.02.2007 11:42...	✓	Нет		руб.		08.02.2007	07.02.2007		Договор к...	Да	1 125,72	
TK0000...	03.02.2007 14:00...	✓	Нет		USD		03.02.2007	06.02.2007		Договор к...	Да	912,35	
TK0000...	10.02.2007 12:00...	✓	Нет		USD		13.02.2007	15.02.2007		Договор к...	Нет	4 149,76	
TK0000...	11.02.2007 0:00:00	✓	Нет		USD		16.02.2007	18.02.2007		Договор И...	Да	1 303,68	Отп...
TK0000...	13.02.2007 12:00...	✓	Нет		USD		18.02.2007	20.02.2007		Договор к...	Да	2 156,05	
TK0000...	13.02.2007 21:18...	✓	Нет		USD		18.02.2007	15.02.2007		Договор п...	Да	4 149,76	
TK0000...	14.02.2007 21:18...	✓	Нет		EUR		19.02.2007	20.02.2007		Договор к...	Да		
TK0000...	15.02.2007 12:00...	✓	Нет		руб.		20.02.2007	21.02.2007		Договор к...	Да	40 083,56	
TK0000...	17.02.2007 12:00...	✓	Нет		руб.		22.02.2007	23.02.2007		Договор к...	Да	17 889,16	
TK0000...	20.02.2007 12:00...	✓	Нет		руб.		22.02.2007	22.02.2007	Дисконтна...	Договор к...	Да	98 398,44	
TK0000...	01.03.2007 12:00...	✓	Нет		руб.		11.03.2007	07.03.2007		Договор к...	Да		
TK0000...	04.03.2007 12:00...	✓	Нет		EUR		09.03.2007	10.03.2007		Договор к...	Да	1 857,42	
TK0000...	04.03.2007 12:00...	✓	Нет		USD		09.03.2007	10.03.2007		Договор к...	Нет		
TK0000...	05.03.2007 12:00...	✓	Нет		USD		10.03.2007	11.03.2007		Договор к...	Нет		
TK0000...	06.03.2007 12:00...	✓	Нет		USD		11.03.2007	12.03.2007		Договор к...	Да		
TK0000...	07.03.2007 12:00...	✓	Нет		USD		12.03.2007	13.03.2007		Договор к...	Да	2 389,68	
TK0000...	08.03.2007 12:00...	✓	Нет		USD		13.03.2007	13.03.2007		Договор к...	Да	2 389,68	
TK0000...	11.03.2007 12:00...	✓	Нет		USD		16.03.2007	17.03.2007		Договор к...	Да		
TK0000...	12.03.2007 0:00:00	✓	Нет		USD		15.03.2007	17.03.2007		Договор к...	Да		
TK0000...	12.03.2007 12:00...	✓	Нет		USD		17.03.2007	18.03.2007		Договор к...	Да		
TK0000...	13.03.2007 12:00...	✓	Нет		USD		18.03.2007	17.03.2007		Договор к...	Да		
TK0000...	17.03.2007 12:00...	✓	Нет		EUR		22.03.2007	23.03.2007		Договор к...	Да		
TK0000...	01.04.2007 13:18...	✓	Нет		руб.		05.04.2007	07.04.2007		Договор к...	Да		
TK0000...	02.04.2007 12:00...	✓	Нет		USD		16.05.2007	04.04.2007		Договор И...	Нет		

Пользоваться этим надо с умом. Если это огромная таблица и там нет индекса по этому полю, то вы можете просто надолго нагрузить сервер базы данных. Поэтому пользуйтесь осторожно!

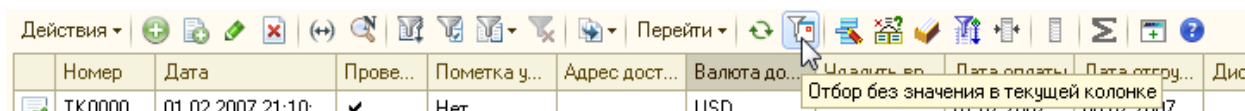
Дополнительные возможности

Можно настраивать режимы работы с табличным полем: иерархический просмотр, многократный выбор и доступность всех элементов отбора. Последний позволяет сделать отбор по реквизитам, недоступным в стандартном диалоге отбора (по умолчанию включены только доступные элементы отбора). Но при этом важно понимать, что отбор/упорядочивание по неиндексированным полям в большой таблице может сильно нагружать сервер.



Команда «Отбор без значения в текущей колонке»

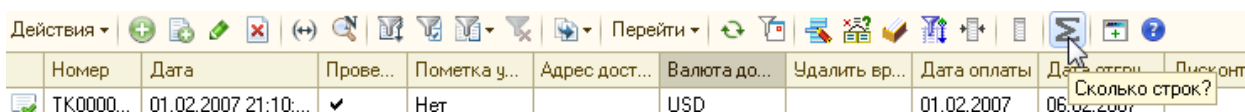
Теперь про кнопку «Отбор без значения в текущей колонке». Это – своего рода антипод той кнопке «Отбор по значению в текущей колонке», к которой все так привыкли.



Например, я нашел какое-то значение в колонке списка, и оно у меня во всех видимых строках отображается. А мне нужны как раз другие значения. Поэтому я нажимаю «Отбор без значения в текущей колонке», и строки с этим значением сразу же все исчезли! Если я хочу еще какие-то значения отбросить, я ее еще раз нажимаю (там уже вид сравнения «Не в списке» будет использоваться).

Команда «Количество строк»

Здесь есть еще одна замечательная вещь – вы можете посмотреть, сколько строк сейчас отображается в динамическом списке с учетом отбора.



Обычно для этого используют команду «Вывести список...», однако при большом количестве строк (больше 1000) ждать результата придется очень долго, причем в процессе

выполнения команды прерывание (Ctrl+Break) не будет работать. А при использовании команды «Количество строк» вы получите результат сразу, без ожидания.