

Авторы книги:

Сайфутдинов Вадим Альбертович
Сотников Анатолий Владимирович

ЯЗЫК ЗАПРОСОВ 1С «ОТ ЗУБОВ»!

ВЫБРАТЬ

<ИмяПоля1> КАК <ПредставлениеПоля1>,
Сумма(<ИмяПоля2>) КАК <ПредставлениеПоля2>
ИЗ

<ИмяТаблицы1> КАК <ПредставлениеТаблицы1>

<ТипСоединения> СОЕДИНЕНИЕ

<ИмяТаблицы2> КАК <ПредставлениеТаблицы2>

ПО <УсловиеСоединениеТаблиц>

ГДЕ

<УсловиеОтбораДанных>

СГРУППИРОВАТЬ ПО

<ИмяПоля1>

УПОРЯДОЧИТЬ ПО

<ИмяПоля1>

ИТОГИ

<ИмяПоля2>

ПО

<ИмяПоля1>



Верный старт в 1С
www.work-1c.ru

Оглавление

Введение.....	4
Глава 1.....	5
1.1 Обзор конфигурации.....	5
1.2. Запрос данных из справочника.....	9
1.3. Метод «Выбрать()».....	14
1.4 Метод «Выгрузить()».....	18
1.5 Параметры запроса.....	19
1.6 Консоль запросов.....	23
1.7 Псевдонимы полей.....	27
1.8 Сортировка.....	28
1.9 Получить первые.....	31
1.10 Без повторяющихся.....	32
1.11 Реальные таблицы. Партии товаров.....	34
1.12 Условия. Между, сравнения.....	35
1.13 Виртуальные таблицы. Остатки.....	38
1.14 Параметры виртуальной таблицы.....	41
1.15 Условия. В иерархии.....	43
Глава 2.....	46
2.1 Обзор конфигурации.....	46
2.2 Таблицы Справочники и Документы.....	51
2.3 Группировки.....	54
2.4 Итоги в запросе.....	59
2.5 Копирование текста запроса.....	62
2.6 Выборка с итогами в отладчике.....	67
2.7 Отчёт с итогами.....	71
2.8 Общие итоги.....	73
2.9 Виртуальная таблица. Регистр накопления обороты.....	75
2.10 Виртуальная таблица. Регистр накопления остатки.....	80
2.11 Виртуальная таблица. Регистр накопления Остатки и обороты.....	80
2.12 Задача на соединение таблиц.....	81
2.13 Соединение таблиц теория.....	84
2.14 Соединение таблиц. Пример.....	87
Глава 3.....	90
3.1 Соединение таблиц. Детальный теоретический пример.....	90
3.2 Объединение таблиц.....	94

3.3 Вложенные запросы.	100
3.4 Функции языка. Строковые.	106
3.5 Функции языка. Работа с датами.	110
3.6 Функции языка. Агрегатные функции.	115
3.7 Функции языка. Ссылки и типы.	118
3.8 Функции языка. Условия. Ссылка.	123
3.9 Функции языка. Условия. В, В иерархии.	124
3.10 Функции языка. Условия. Есть NULL.	126
3.11 Функции языка. Условия. Выбор.	127
3.12 Функции языка. Условия. Подобно.	131
3.13 Функции языка. Условия. И, ИЛИ, Не.	134
Глава 4.	136
4.1 Параметры виртуальных таблиц на примере условия запроса. Временные таблицы.	136
4.2 Менеджер Временных Таблиц.	142
4.3 Уничтожение временных таблиц.	155
4.4 Пакет запросов.	157
Глава 5.	173
5.1. Оперативный учет.	173
5.1.1. Ввод в эксплуатацию.	173
5.1.2 Индексирование.	196
5.1.3 Выбытие оборудования.	198
5.1.4 Метод дополнения.	209
5.2 Бухгалтерский учет.	214
5.2.1 Бухгалтерский учет. Списание комплектов и комплектующих.	214
5.2.2 Бухгалтерский учет. Таблицы.	237
5.3 Параметры виртуальных таблиц регистров расчета.	247
5.4 Контроль разрешения прав доступа на уровне записей. «Разрешенные».	250
5.5 Блокировка данных для последующего изменения.	250

Введение

Здравствуйте, уважаемые читатели!

Мы, авторы проекта «Верный старт в 1С», Сайфутдинов Вадим и Сотников Анатолий, рады представить вам нашу новую книгу «Язык запросов 1С “ОТ ЗУБОВ”».

Основываясь на своем многолетнем опыте в сфере 1С, в том числе, практике внедрений конфигураций и преподавания, мы сделали вывод о необходимости подготовить качественный и подробный материал по данному вопросу для начинающих 1С-программистов.

Книга состоит из 5 глав, каждая из которых последовательно погружает вас во все более сложные и интересные задачи, которые можно решить с помощью такого мощного инструмента получения данных, как язык запросов. Текст книги на всем протяжении сопровождается подробными инструкциями и наглядными иллюстрациями, однако, для более продуктивной работы и получения знаний «от зубов», мы все же рекомендуем вам закреплять полученные знания самостоятельным выполнением практических заданий.

Итак, мы начинаем!

Глава 1.

1.1 Обзор конфигурации.

Для начала рассмотрим рабочую среду, в которой нам предстоит работать. Это учебная платформа, которую вы можете скачать по ссылке <https://yadi.sk/d/VouBGzDuGry5PQ>. Для изучения данной главы также необходимо скачать информационную базу <https://yadi.sk/d/VFiwWkk5R0T18w>.

Откроем информационную базу в режиме «Конфигуратор». Посмотрим состав объектов (Рисунок 1).

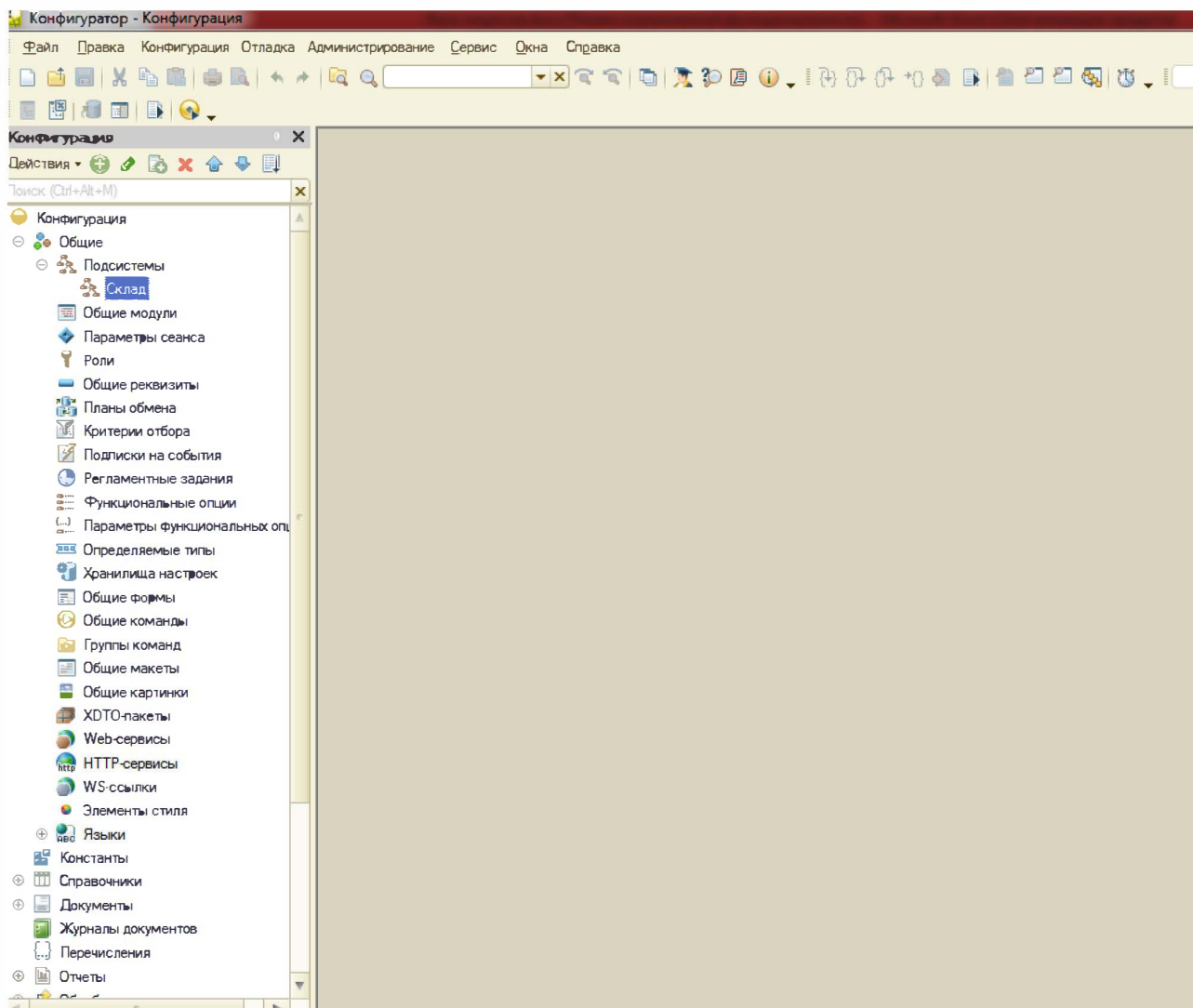


Рисунок 1

В конфигурации имеется одна подсистема («Склад») и несколько справочников (Рисунок 2):

1. «Номенклатура», с единственным реквизитом «Единица измерения»;
2. «Единицы измерения» – без реквизитов;
3. «Склады» – так же без реквизитов.

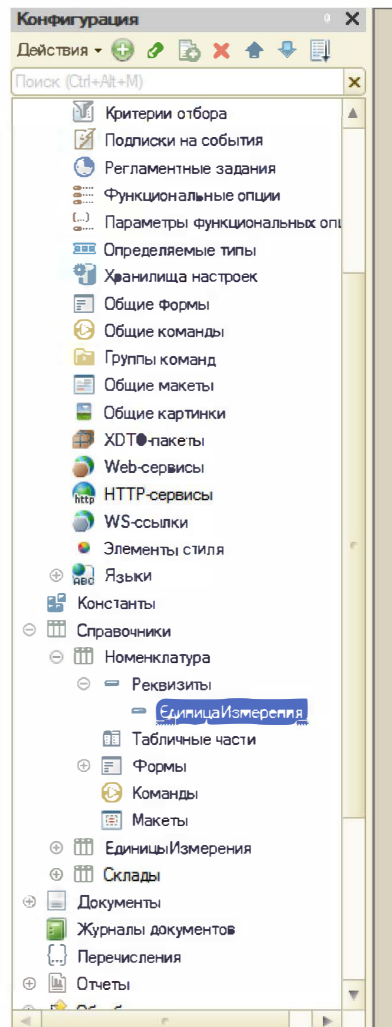


Рисунок 2

Справочник «Единицы измерения» имеет predetermined (added by the developer in the «Configurator» mode) element – «Штука». Увидеть его можно перейдя на закладку «Прочее» окна свойств справочника и нажав кнопку «Предопределенные» (Рисунок 3).

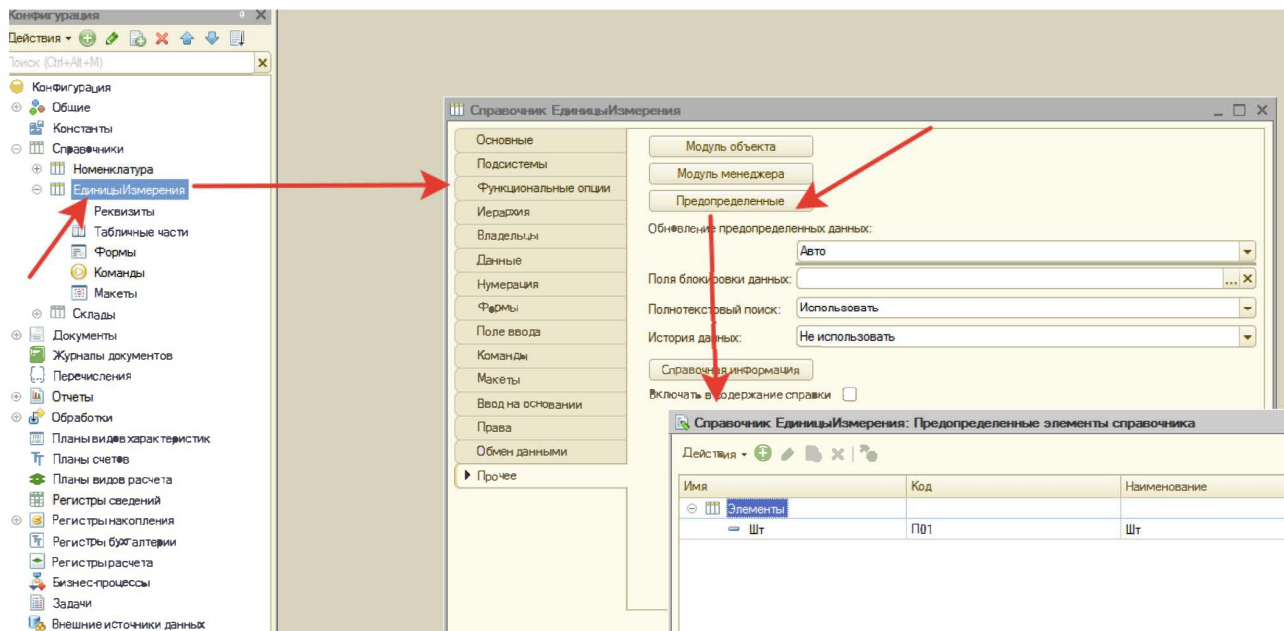


Рисунок 3

Из документов в данной базе есть только «Оприходование товаров» с реквизитом «Склад» и табличной частью «Товары». Табличная часть, в свою очередь, содержит реквизиты: «Номенклатура», «Количество», «Сумма» (Рисунок 4).

Этот документ делает движения по регистру накопления «Партии товаров». Регистр состоит из измерений: «Склад», «Номенклатура», «Партия» и ресурсов: «Количество», «Сумма». Измерение «Партия» – это сам документ оприходования.

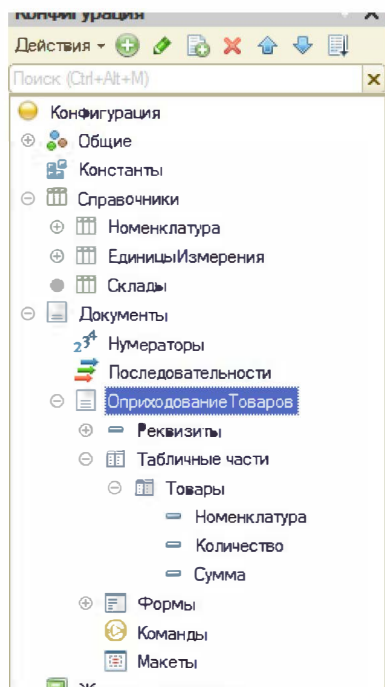


Рисунок 4

В конфигурации имеется обработка «Консоль запросов» – это основной инструмент, с которым нам предстоит работать в данном курсе. С ней мы познакомимся позже, для начала рассмотрим отчеты.

В данной конфигурации имеется один отчет – «Отчет по номенклатуре» с макетом табличного документа, при помощи которого отчет выводится на экран (Рисунок 5).

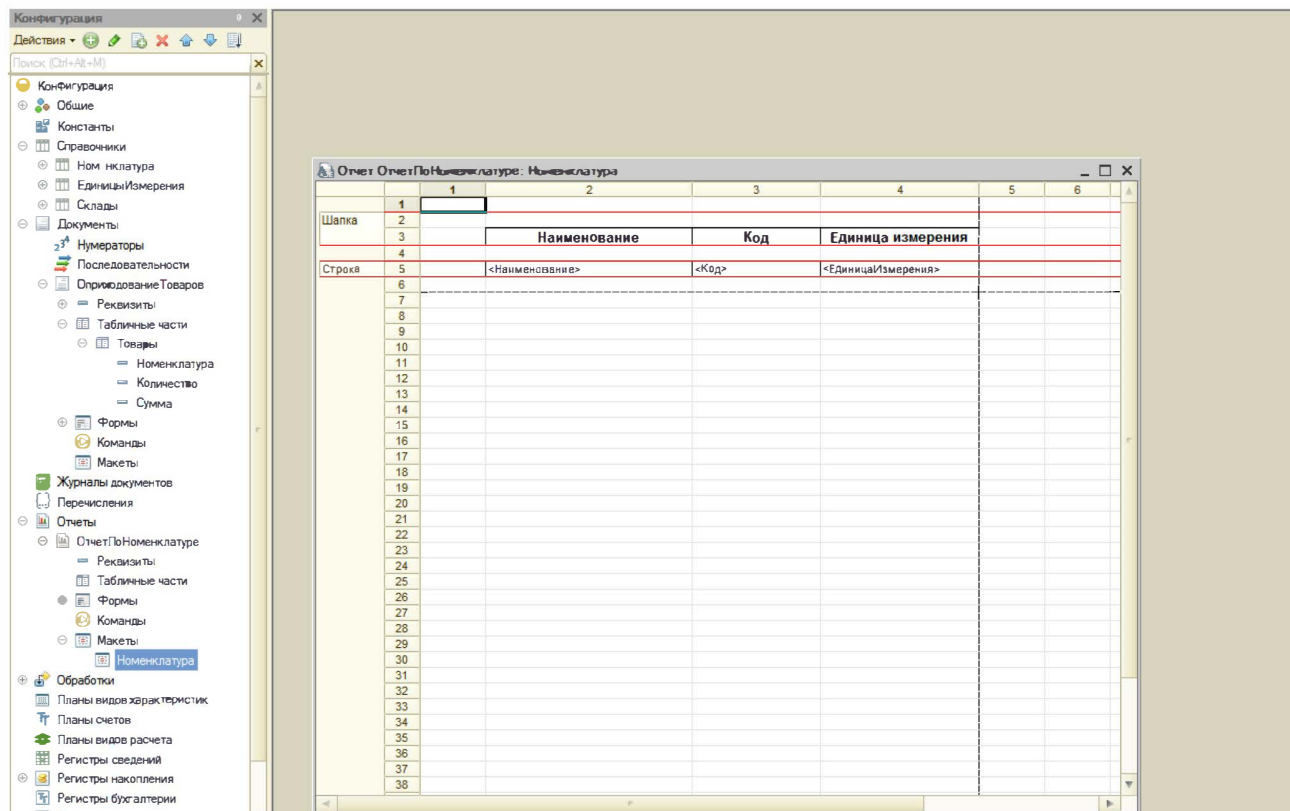


Рисунок 5

Весь код по формированию отчета находится в модуле формы (Рисунок 6).

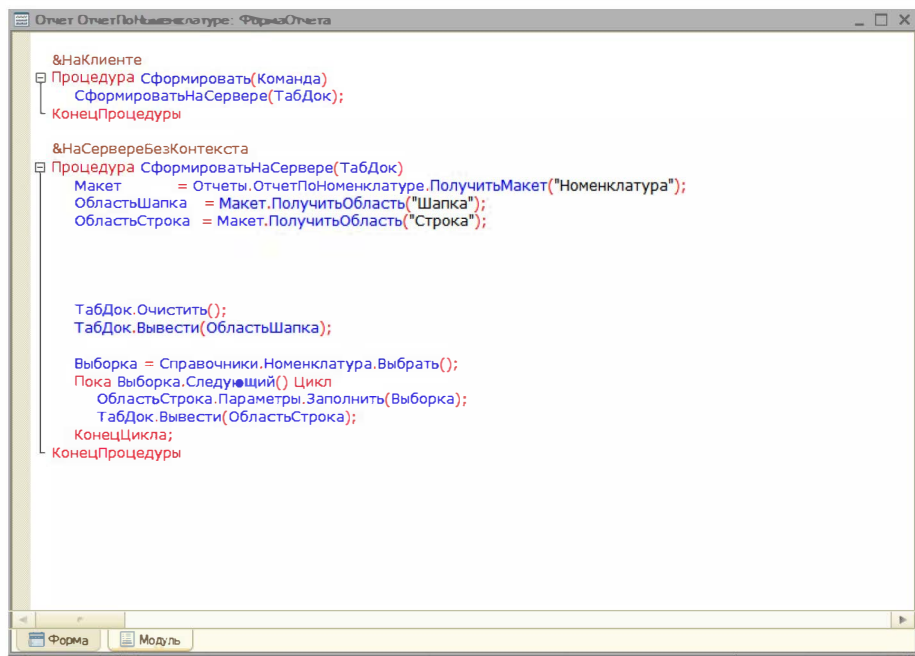


Рисунок 6

Перейдем в режим «1С:Предприятие» и посмотрим на нашу конфигурацию (Рисунок 7).

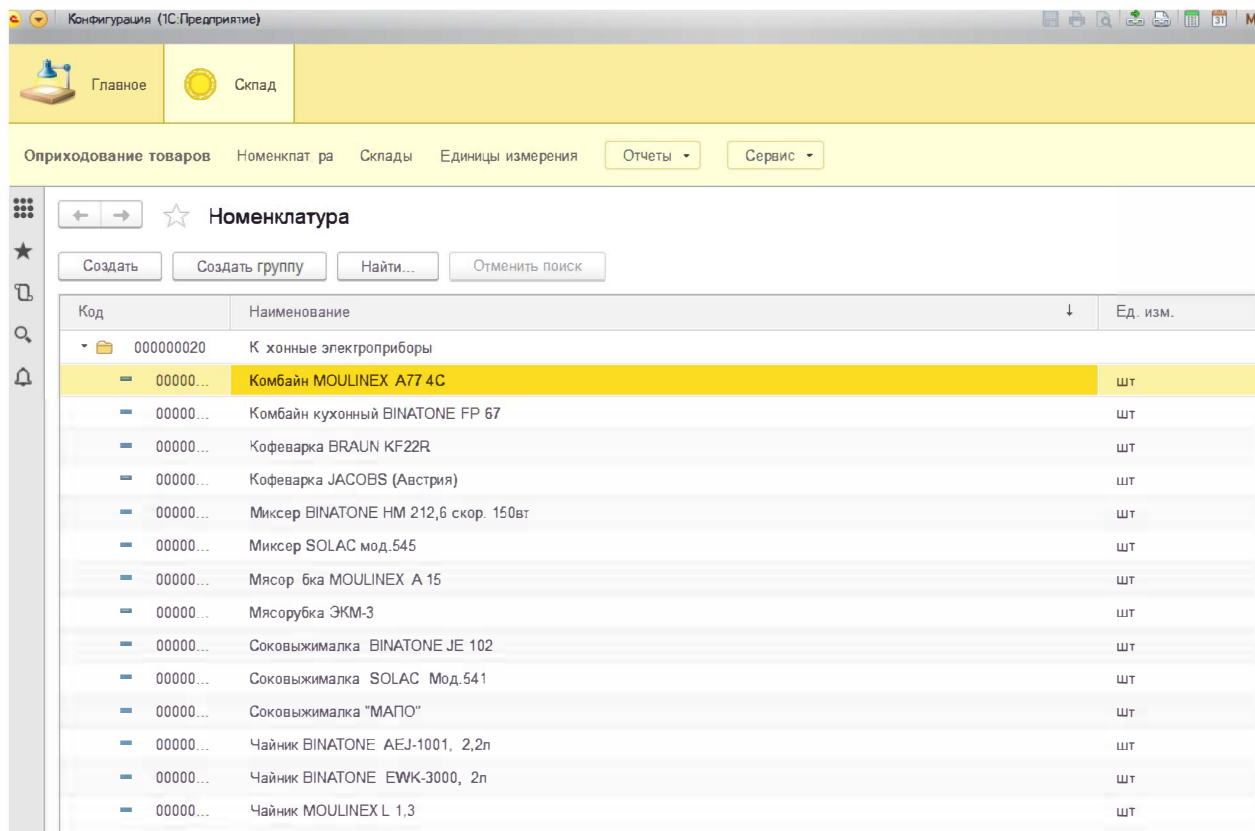


Рисунок 7

На панели разделов можно увидеть подсистему «Склад», а в ней подразделы: «Оприходование товаров», «Номенклатура», «Склады», «Единицы измерения», «Отчеты» и «Сервис». В подразделе «Отчеты» находится «Отчет по номенклатуре», в подразделе «Сервис» – обработка «Консоль запросов».

По мере прохождения курса наша конфигурация будет заметно усложняться, но об этом в следующих уроках.

1.2. Запрос данных из справочника.

В языке 1С сосуществуют 2 модели получения данных из базы: объектная модель и табличная модель. Объектная модель представляет данные в виде объектов, то есть мы работаем с данными, как с программными объектами. Пример объектной модели получения данных представлен на рисунке 8.

```

Выборка = Справочники.Номенклатура.Выбрать();
Пока Выборка.Следующий() Цикл
    ОбластьСтрока.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(ОбластьСтрока);
КонецЦикла;

```

Рисунок 8

Мы обращаемся к некому объекту «Справочники.Номенклатура», к его методу Выбрать(), который возвращает нам данные, помещая их в переменную «Выборка».

Представленный выше код поначалу может показаться трудным и непонятным, пусть это вас не пугает. Абстрагируйтесь от деталей и поймите главное: в данном случае мы получаем данные при помощи объекта «Справочники.Номенклатура». Однако мы должны понимать, что, в сущности, все данные в 1С представлены в виде таблиц, поэтому когда мы обращаемся к вышеуказанному объекту для получения данных, система незаметно для нас обращается с запросом к базе данных. Таким образом, скрыто или явно, для получения данных из базы в 1С всегда используется запрос.

Перейдем к рассмотрению второй модели получения данных – табличной. Составим наш первый запрос.

Для составления запроса необходимо объявить новый объект, который так и называется «Запрос». Запрос создается при помощи конструктора с использованием ключевого слова «Новый». У объекта типа «Запрос» есть свойство «Текст», в которое необходимо поместить текст нашего запроса.

Для создания запросов в платформе «1С:Предприятие» существует замечательный инструмент «Конструктор запросов». Самый простой способ его вызова – контекстное меню в редакторе кода (Рисунок 9).

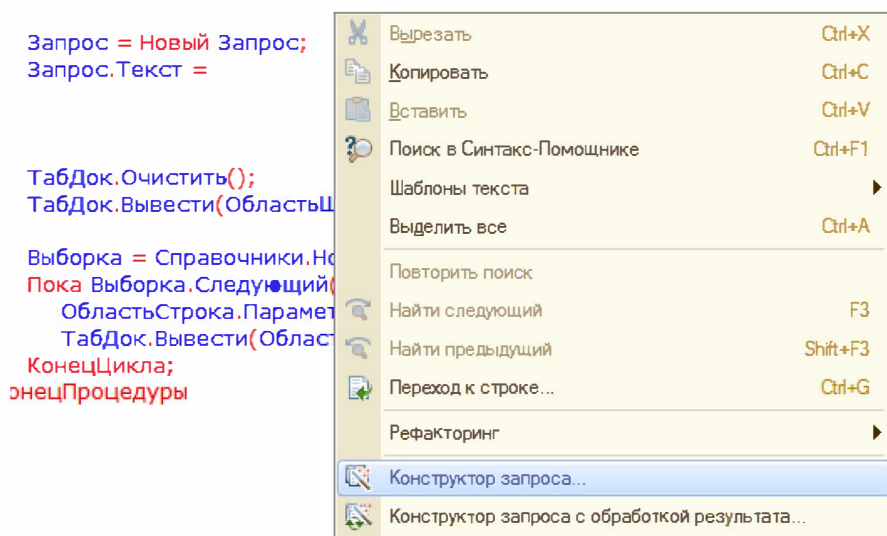


Рисунок 9

Система спросит нас о необходимости создания нового запроса, отвечаем «Да» (Рисунок 10).

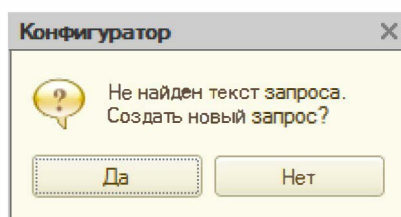


Рисунок 10

Далее перед нами откроется конструктор запроса:

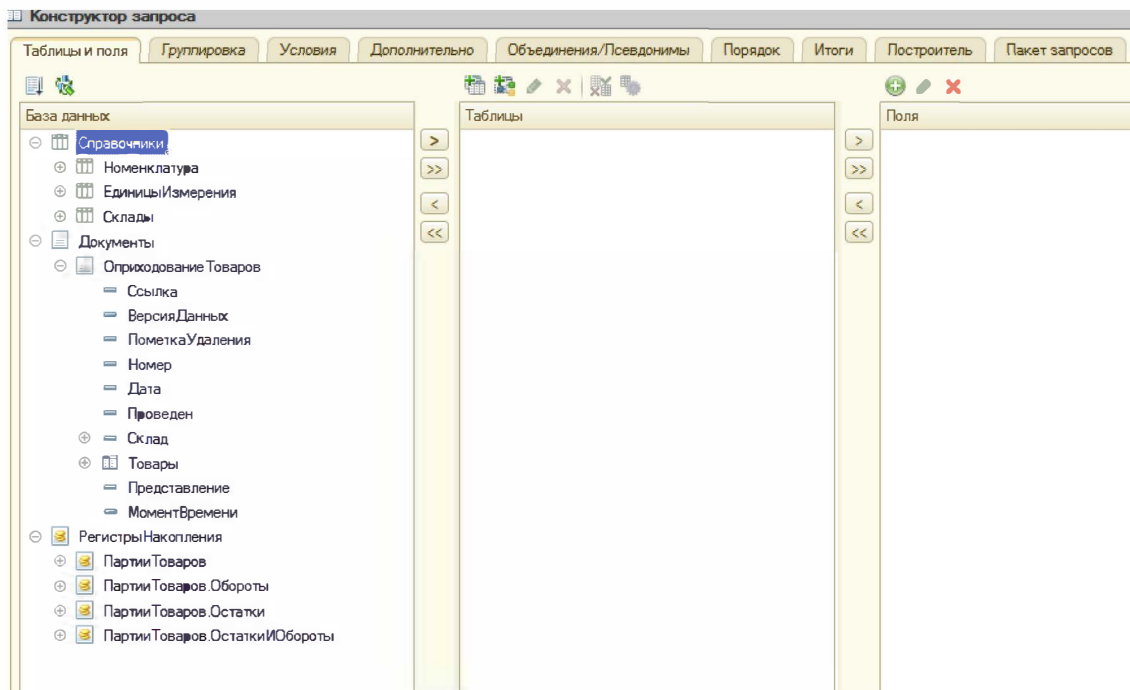



Рисунок 11

На закладке «Таблицы и поля» находятся перечень всех таблиц, существующих в нашей базе данных, они расположены в левой трети окна. Далее нам необходимо выбрать таблицу, из которой мы хотим получить данные. Для этого можно дважды щелкнуть левой кнопкой мыши по таблице или нажать кнопку со стрелочкой: , также можно воспользоваться перетаскиванием в центральную треть окна. Итак, выбираем справочник «Номенклатура». Развернем его, нажав на значок крестика слева от таблицы, и увидим состав объекта, его реквизиты или, в терминах базы данных, «поля» (Рисунок 12).

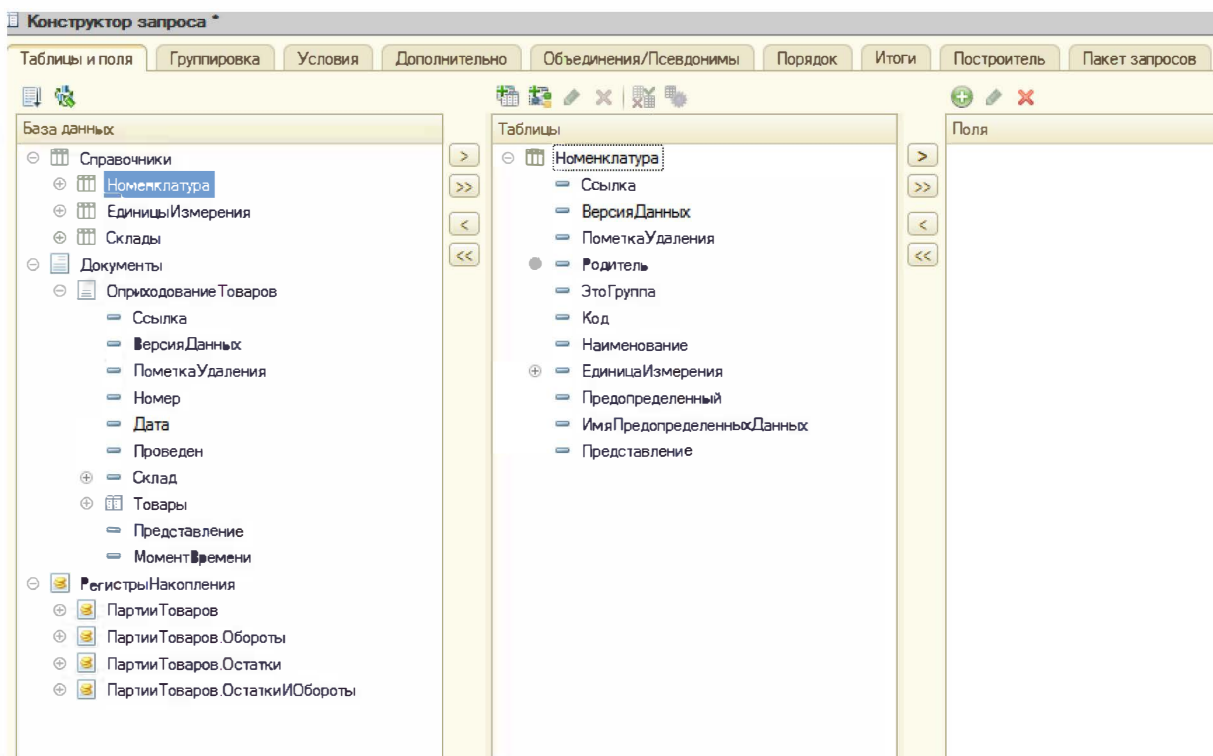


Рисунок 12

Теперь необходимо выбрать те из них, которые нам потребуются для нашего примера, в частности, такие поля как: наименование, код и единица измерения. Поля выбираются так же, как и таблицы, только выбор, на этот раз, необходимо делать в центральной трети окна, которая называется «Таблицы». Туда мы поместили выбранную таблицу. Выбранные поля отображаются в правой трети окна, которая так и называется «Поля» (Рисунок 13).

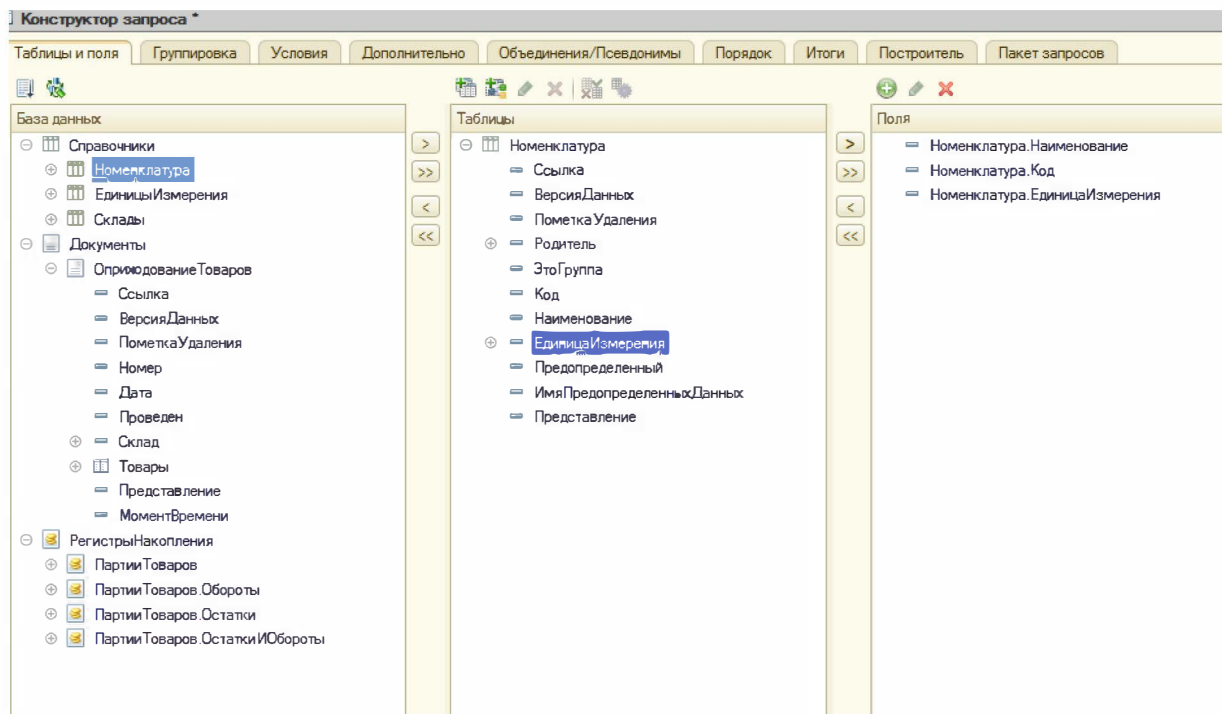


Рисунок 13

Нажимаем кнопку «ОК» и получаем сформированный текст запроса (Рисунок 14).

```
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
| Номенклатура.Наименование,
| Номенклатура.Код,
| Номенклатура.ЕдиницаИзмерения
| ИЗ
| Справочник.Номенклатура КАК Номенклатура";
```

Рисунок 14

Обратите внимание, что текст запроса представлен в виде строки. Все строковые значения в языке 1С обрамляются двойными кавычками. Для улучшения восприятия текстовые строки можно переносить, для этого используется символ «|» – так платформа понимает, что текстовая строка продолжается. Каждый оператор завершается точкой с запятой. Конструктор запроса сгенерировал запрос в виде строки. Эта строка присвоена свойству «Текст» переменной «Запрос». Поэтому оператор присвоения завершается точкой с запятой.

Далее получим выборку из нашего запроса. Для этого нам необходимо написать:

```

Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
| Номенклатура.Наименование,
| Номенк атура.Код,
| Номенк атура.ЕдиницаИзмерения
|ИЗ
| Справочник.Номенк атура КАК Номенк атура";


ТабДок.Очистить();
ТабДок.Вывести(ОбластьШапка);

Выборка = Запрос.Выполнить().Выбрать();

```

Рисунок 15

Ключевой является последняя строка, в которой мы получаем выборку не объектным методом, а табличным, при помощи запроса.

Далее убедимся, что наш отчет получает те же данные, что и раньше. Для этого обновим конфигурацию базы данных (сохраним проведенные нами изменения в конфигурации), нажав кнопку  на панели инструментов. Запустим конфигурацию в режиме «1С:Предприятие» и сформируем отчет (Рисунок 16).

← → ☆ Отчет по номенклатуре

Сформировать

Таб док:

Наименование	Код	Единица измерения
Кухонные электроприборы	000000020	
Комбайн кухонный BINATONE FP 67	000000021	шт
Кофеварка BRAUN KF22R	000000022	шт
Кофеварка JACOBS (Австрия)	000000023	шт
Миксер BINATONE HM 212,6 скор. 150вт	000000024	шт
Холодильники, морозильные камеры	000000025	
BOSCH	000000026	шт
МИНСК-АТЛАНТ 126	000000027	шт
МИНСК-АТЛАНТ 215	000000028	шт
СТИНОЛ 101	000000029	шт
СТИНОЛ 103	000000030	шт
Вентиляторы, пылесосы, кондиционеры	000000031	
Вентилятор BINATONE ALPINE 160вт, напольный	000000032	шт
Вентилятор JIPONIC (Тайв.),	000000033	шт
Вентилятор настольный	000000034	шт
Комбайн MOULINEX A77 4С	000000046	шт
Миксер SOLAC мод.545	000000047	шт
Мясорубка ЭКМ-3	000000048	шт
Вентилятор оконный	000000049	шт
Вентилятор ОРБИТА, STERLING, ЯП.	000000050	шт
Телевизоры	000000056	
Телевизор "JVC"	000000057	шт
Кондиционер ELEKTA	000000058	шт
Кондиционер FIRMSTAR 12M	000000059	шт
Кондиционер БК-2300	000000060	шт
Пылесос "Омега" 1250вт	000000061	шт
Пылесос "Электросила"	000000062	шт
Пылесос "Энергия-SANYO"	000000063	шт
Чайник BINATONE EWK-3000, 2л	000000111	шт
Чайник BINATONE AEJ-1001, 2,2л	000000113	шт
Мясорубка MOULINEX A 15	000000115	шт
Соковыжималка BINATONE JE 102	000000116	шт

Рисунок 16

Как мы видим, результат не изменился.

1.3. Метод «Выбрать()».


Для дальнейшего изучения языка запросов нам будет необходимо воспользоваться отладчиком. Это очень удобный, нужный и полезный инструмент любого разработчика. Для того чтобы им воспользоваться нам нужно запустить конфигурацию в особом режиме – режиме отладки. Для запуска конфигурации в таком режиме достаточно в режиме «Конфигуратор» нажать клавишу F5 или на панели инструментов нажать кнопку . Итак, запуск конфигурации в режиме отладки дает возможность пошагово посмотреть выполнение нашей программы. Для того чтобы «вклиниться» в ход выполнения программы существуют точки останова. Для того чтобы установить точку останова необходимо на нужной строке программного кода нажать клавишу F9, повторное нажатие этой клавиши удаляет точку останова. Точка останова будет обозначена специальной пиктограммой в виде красного круга (Рисунок 17).



Рисунок 17

Теперь, в режиме отладки, программа, дойдя до точки останова, приостановит свое выполнение в ожидании наших дальнейших действий. О том, на какой строке исполняемого кода находится программа, будет информировать специальный указатель в виде стрелки слева (Рисунок 18).

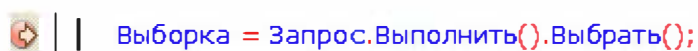


Рисунок 18

Какие же дальнейшие действия мы можем выполнить?

Их несколько:

1. «Шагнуть в» (клавиша F11) – выполняется один оператор, но если он содержит обращение к процедуре или функции, мы войдем в нее.
2. «Шагнуть через» (клавиша F10) – выполняется один оператор, если он содержит обращение к процедуре или функции, мы НЕ войдем в нее.
3. «Шагнуть из» (сочетание клавиш Shift+ F11) – выходим из текущей процедуры или функции, в которой находимся, в ту процедуру или функцию, которая её вызвала. Если же вызов был из пользовательского режима, то мы вернемся в пользовательский режим.
4. «Идти до курсора» (сочетание клавиш Shift+ F10) – выполнение кода продолжается до той строки, на которой находится курсор, затем происходит останов, как на точке останова.
5. «Продолжить отладку» (клавиша F5) – выполнение программы будет продолжено до следующей точки останова, а если такой нет, то до конца.

6. «Перезапустить» (сочетание клавиш Ctrl+Shift+ F5) – отладка будет полностью перезапущена.

7. «Завершить» (сочетание клавиш Shift+ F5) – отладка будет завершена, то есть программа закроется, но конфигуратор останется запущенным.

Итак, запускаем конфигурацию в режиме отладки и ставим точку останова на строке (Рисунок 19).



Рисунок 19

Открываем наш отчет: «Склад» -> «Отчеты» -> «Отчет по номенклатуре», далее нажимаем на кнопку «Сформировать». Теперь мы остановились на заданной точке останова. Все предыдущие «мучения» с запуском отладки, точками останова и ознакомлением действий в отладке были проделаны ради возможности исследовать наши программные объекты: «Запрос» и «Выборка». Что ж, приступим к этому увлекательному занятию.

Для того чтобы просмотреть содержимое переменной, необходимо установить на ней курсор и нажать комбинацию клавиш Shift+F9. Посмотрим на переменную «Запрос» (Рисунок 20).



Рисунок 20

Как видим, переменная состоит из свойств «МенеджерВременныхТаблиц», «Параметры» и «Текст». Свойство «Текст» содержит текст нашего запроса, который мы создавали при помощи конструктора. Убедиться в этом можно, выделив его и нажав кнопку «Показать в отдельном окне» или нажав клавишу (Рисунок 21-22).

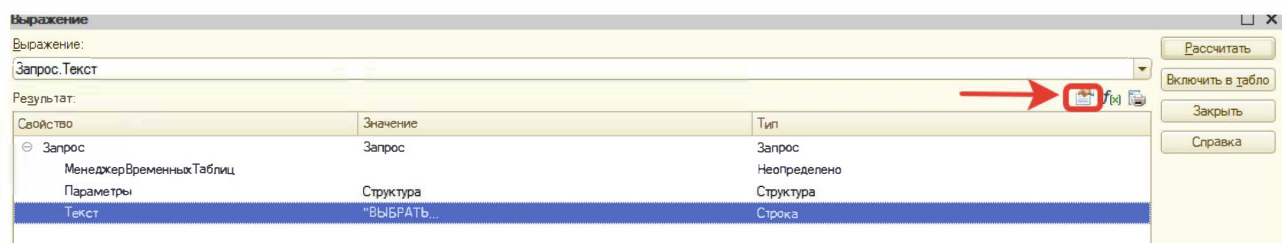


Рисунок 21

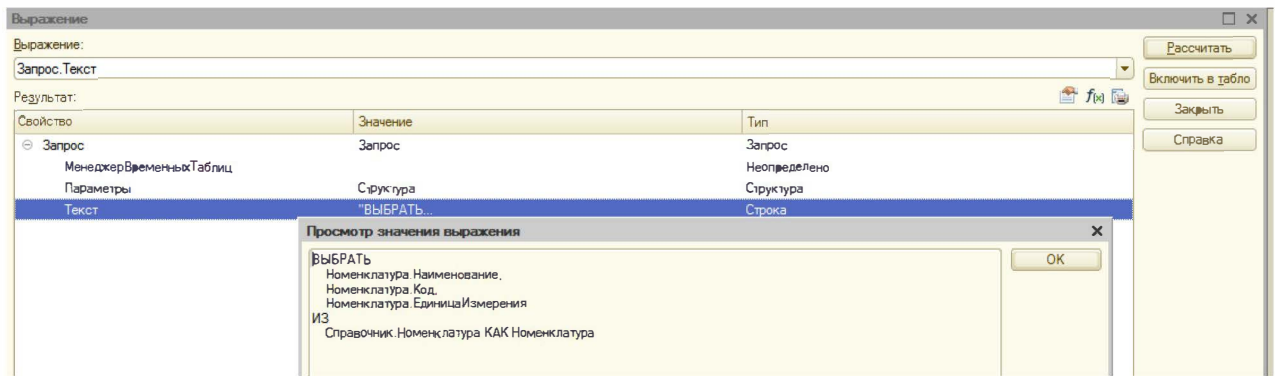


Рисунок 22

Вы можете задать логичный вопрос: «А зачем нам смотреть текст запроса в отладчике, он же прописан в программном коде?». Но это не всегда так. В типовых конфигурациях фирмы 1С таких как: «Управление торговлей» или «Бухгалтерия предприятия» запросы все чаще формируются динамически, то есть из разных кусков в зависимости от условий. Увидеть конечный текст запроса нам как раз и поможет отладчик.

Далее посмотрим, что получится при вызове метода запроса «Выполнить()».

Выделяем выражение «Запрос.Выполнить()» и нажимаем Shift+F9. Видим, что метод возвращает тип данных «РезультатЗапроса». В нем есть колонки, которые содержат информацию о типе значения и ширине, но самих данных в этом объекте мы не видим (Рисунок 23).

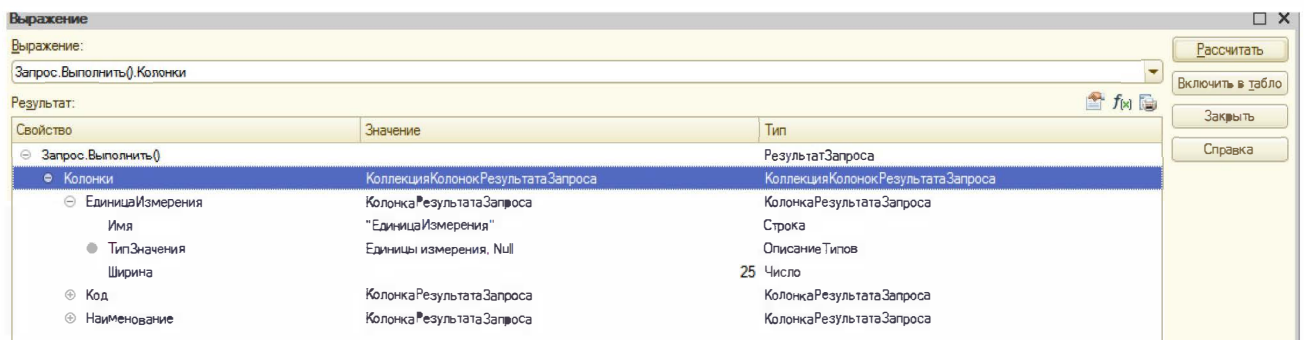


Рисунок 23

Для получения данных нам необходимо обратиться к методу «Выбрать()» результата запроса. Этот метод возвращает выборку из результата запроса, из которой мы можем получить сами данные, обходя выборку в цикле.

Здесь внесем ясность в то, что такое методы, а что такое свойства какого-либо объекта. Свойством принято считать элемент внутренней структуры объекта, содержащий данные (сведения об объекте).

Например, мы имеем объект «Автомобиль», у которого есть свойства: «Масса», «МаксимальнаяСкорость», «Цвет» и другие. Все эти свойства характеризуют сам объект и являются составными частями этого объекта. К этим свойствам мы можем обращаться для чтения или для записи. Обращения к свойствам происходит через точку:

Автомобиль.Масса= 1200;

Автомобиль.Цвет = Желтый;

МахСкорость = Автомобиль.МаксимальнаяСкорость;

Методы же представляют собой некие внутренние процедуры или функции, призванные изменять и обрабатывать данные. Обращение к методам происходит также через точку, но в конце обязательно дописываются скобки. Иногда в скобках помещают параметры – это некие входные данные для обработки. Параметр может быть один или несколько, в таком случае они перечисляются через запятую. Если же параметров нет, то скобки остаются пустые.

Вернемся к методу «Выбрать()» результата запроса. Он работает как функция, то есть возвращает нам некое значение, а именно, выборку из результата запроса, которой мы даем имя «Выборка». Это происходит в строке программного кода, представленного на рисунке 24.

```
Выборка = Запрос.Выполнить().Выбрать();
```

Рисунок 24

Другими словами, переменной «Выборка» мы присваиваем значение выборки из результата запроса, которое возвращает нам метод «Выбрать()» результата запроса. Убедимся в этом, нажав F10, находясь на точке останова в момент отладки.

Теперь установим курсор на переменной «Выборка» и нажмем Shift+F9 (Рисунок 25).

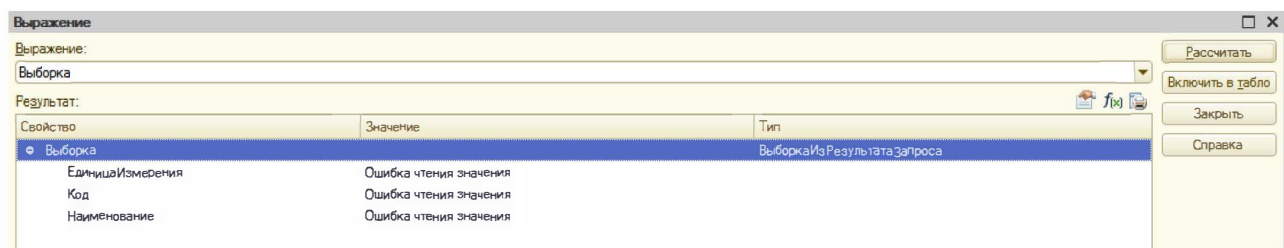


Рисунок 25

Здесь мы видим поля, выбранные в запросе, но снова не видим данных. Это происходит по причине того, что наша выборка является коллекцией элементов, которую нужно обходить при помощи метода «Следующий()». Данный метод возвращает значение «Истина», если следующий элемент считан, и «Ложь», если считывать больше нечего. Поэтому метод «Следующий()» выполняет несколько функций в нашем коде:

1. Считывает следующий элемент выборки.
2. Является условием выполнения цикла.

Продолжим отладку, нажмем F11 и снова посмотрим на переменную «Выборка» (Рисунок 26):

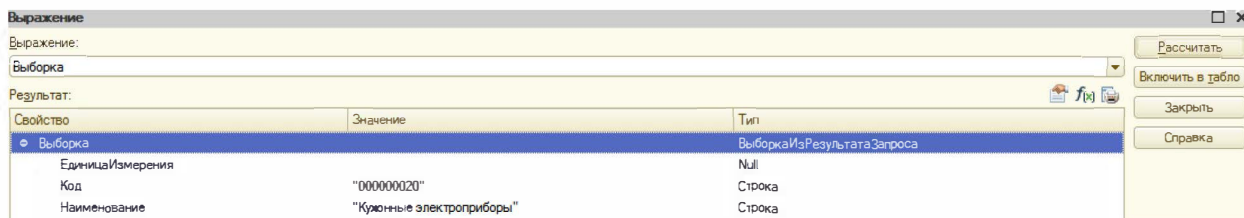


Рисунок 26

И вот уже сейчас мы видим первый элемент нашей выборки. Таким образом, в этом цикле последовательно выполняется обход всех данных, которые получены нашим запросом.

1.4 Метод «Выгрузить()».

Результат запроса, помимо прочего, имеет метод «Выгрузить()», который в отличие от метода «Выбрать()», возвращает таблицу значений. ТаблицаЗначений – тип данных в 1С, отличающийся от выборки, прежде всего тем, что полностью размещается в оперативной памяти и его можно просмотреть целиком, в то время как выборку можно просмотреть только поэлементно.

Давайте же посмотрим, как он работает.

Точка останова у нас по-прежнему установлена на строке (Рисунок 27.)

`Выборка = Запрос.Выполнить().Выбрать();`

Рисунок 27

Конфигурация запущена в режиме отладки. В пользовательском режиме нажимаем на кнопку отчета «Сформировать». Далее останавливаемся на нужной строке. Выделяем выражение «Запрос.Выполнить()», нажимаем сочетание клавиш Shift+F9 и прямо в окне вычисления выражения дописываем «.Выгрузить()» и нажимаем «Рассчитать» (Рисунок 28).

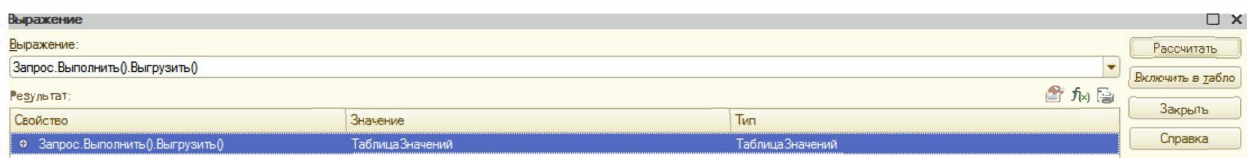


Рисунок 28

Мы получили таблицу значений и теперь, выделив ее, нажимаем F2 для просмотра ее содержимого (Рисунок 29).

Таблица значений					
Количество элементов: <input type="text" value="36"/>					
Индекс	Значение элемента	Тип элемента	Наименование	Код	ЕдиницаИзм
0	Строка Таблицы значений	Строка Таблицы значений	"Кухонные электроприборы"	"000000020"	
1	Строка Таблицы значений	Строка Таблицы значений	"Комбайн кухонный BINATONE F...	"000000021"	шт
2	Строка Таблицы значений	Строка Таблицы значений	"Кофеварка BRAUN KF22R"	"000000022"	шт
3	Строка Таблицы значений	Строка Таблицы значений	"Кофеварка JACOBS (Австрия)"	"000000023"	шт
4	Строка Таблицы значений	Строка Таблицы значений	"Миксер BINATONE HM 212.6 ско...	"000000024"	шт
5	Строка Таблицы значений	Строка Таблицы значений	"Холодильники, морозильные ка...	"000000025"	
6	Строка Таблицы значений	Строка Таблицы значений	"BOSCH"	"000000026"	шт
7	Строка Таблицы значений	Строка Таблицы значений	"МИНСК-АТЛАНТ 126"	"000000027"	шт
8	Строка Таблицы значений	Строка Таблицы значений	"МИНСК-АТЛАНТ 215"	"000000028"	шт
9	Строка Таблицы значений	Строка Таблицы значений	"СТИНОЛ 101"	"000000029"	шт
10	Строка Таблицы значений	Строка Таблицы значений	"СТИНОЛ 103"	"000000030"	шт
11	Строка Таблицы значений	Строка Таблицы значений	"Вентиляторы, пылесосы, конди...	"000000031"	
12	Строка Таблицы значений	Строка Таблицы значений	"Вентилятор BINATONE ALPINE 1...	"000000032"	шт
13	Строка Таблицы значений	Строка Таблицы значений	"Вентилятор JIPONIC (Таив.),"	"000000033"	шт
14	Строка Таблицы значений	Строка Таблицы значений	"Вентилятор настольный"	"000000034"	шт
15	Строка Таблицы значений	Строка Таблицы значений	"Комбайн MOULINEX A77 4C"	"000000046"	шт
16	Строка Таблицы значений	Строка Таблицы значений	"Миксер SOLAC мод.545"	"000000047"	шт
17	Строка Таблицы значений	Строка Таблицы значений	"Мясорубка ЭКМ-3"	"000000048"	шт
18	Строка Таблицы значений	Строка Таблицы значений	"Вентилятор оконный"	"000000049"	шт
19	Строка Таблицы значений	Строка Таблицы значений	"Вентилятор ОРБИТА, STERLING...	"000000050"	шт
20	Строка Таблицы значений	Строка Таблицы значений	"Телевизоры"	"000000056"	
21	Строка Таблицы значений	Строка Таблицы значений	"Телевизор "JVC""	"000000057"	шт
22	Строка Таблицы значений	Строка Таблицы значений	"Кондиционер ELEKTA"	"000000058"	шт
23	Строка Таблицы значений	Строка Таблицы значений	"Кондиционер FIRMSTAR 12M"	"000000059"	шт
24	Строка Таблицы значений	Строка Таблицы значений	"Кондиционер БК-2300"	"000000060"	шт
25	Строка Таблицы значений	Строка Таблицы значений	"Пылесос "Омега" 1250вт"	"000000061"	шт
26	Строка Таблицы значений	Строка Таблицы значений	"Пылесос "Электросила""	"000000062"	шт
27	Строка Таблицы значений	Строка Таблицы значений	"Пылесос "Энергия-SANYO""	"000000063"	шт
28	Строка Таблицы значений	Строка Таблицы значений	"Чайник BINATONE EWK-3000, 2...	"000000111"	шт
29	Строка Таблицы значений	Строка Таблицы значений	"Чайник BINATONE AEJ-1001, 2...	"000000113"	шт
30	Строка Таблицы значений	Строка Таблицы значений	"Мясорубка MOULINEX A 15"	"000000115"	шт
31	Строка Таблицы значений	Строка Таблицы значений	"Соковыжималка BINATONE JE ..."	"000000116"	шт
32	Строка Таблицы значений	Строка Таблицы значений	"Чайник MOULINEX L 1,3"	"000000117"	шт
33	Строка Таблицы значений	Строка Таблицы значений	"Телевизор "SHARP""	"000000120"	шт
34	Строка Таблицы значений	Строка Таблицы значений	"Соковыжималка SOLAC Мод.54...	"000000122"	шт
35	Строка Таблицы значений	Строка Таблицы значений	"Соковыжималка "МАПО""	"000000123"	шт

Рисунок 29

Вот так мы можем увидеть все данные, полученные из запроса, что в некоторых случаях бывает необходимо.

1.5 Параметры запроса.

Предположим, что нам нужно выбрать не весь справочник «Номенклатура», а лишь элементы, отвечающие определенным условиям. На практике, чаще всего, выбирать все элементы нет необходимости.

Для того чтобы ограничить выборку, в конструкторе запроса есть вкладка «Условия». На этой вкладке мы можем определить те условия, по которым будут отбираться данные. Например, нам необходимо выбрать только те элементы, у которых единица измерения штука. Установив курсор на текст запроса, вызываем контекстное меню правой кнопкой мыши и выбираем пункт «Конструктор запроса». Далее переходим на закладку «Условия», разворачиваем структуру справочника и выбираем реквизит «ЕдиницаИзмерения», дважды щелкнув по нему левой кнопкой мыши. Конструктор запроса автоматически сформировал условие (Рисунок 30).

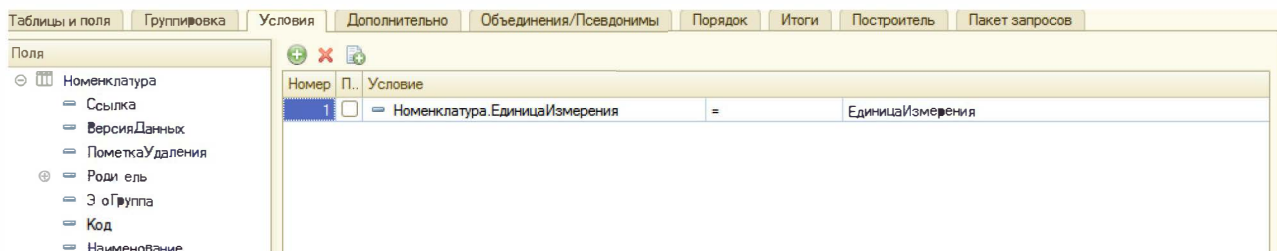


Рисунок 30

Мы можем переопределить это условие самостоятельно. Самый простой способ сделать это – изменить вид сравнения. Если щелкнуть по полю с условием, то мы увидим выпадающий список, в котором присутствуют все возможные виды сравнения (Рисунок 31).

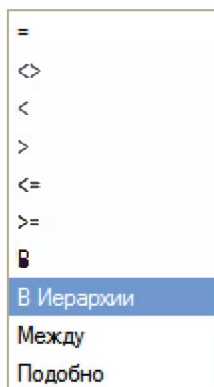


Рисунок 31

Условия могут быть довольно сложными и их можно редактировать, для этого достаточно установить флажок «Произвольное» (Рисунок 32).

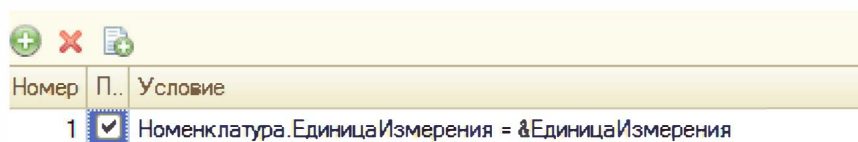


Рисунок 32

Но сейчас нас это не интересует. В данный момент мы задали условие, что единица измерения из номенклатуры должна равняться некой «ЕдиницаИзмерения». Что же это? Это и есть наш параметр, который будет передан в запрос до начала его выполнения.

Завершаем редактирование запроса и нажимаем на «ОК». Посмотрим, как изменился запрос (Рисунок 33).

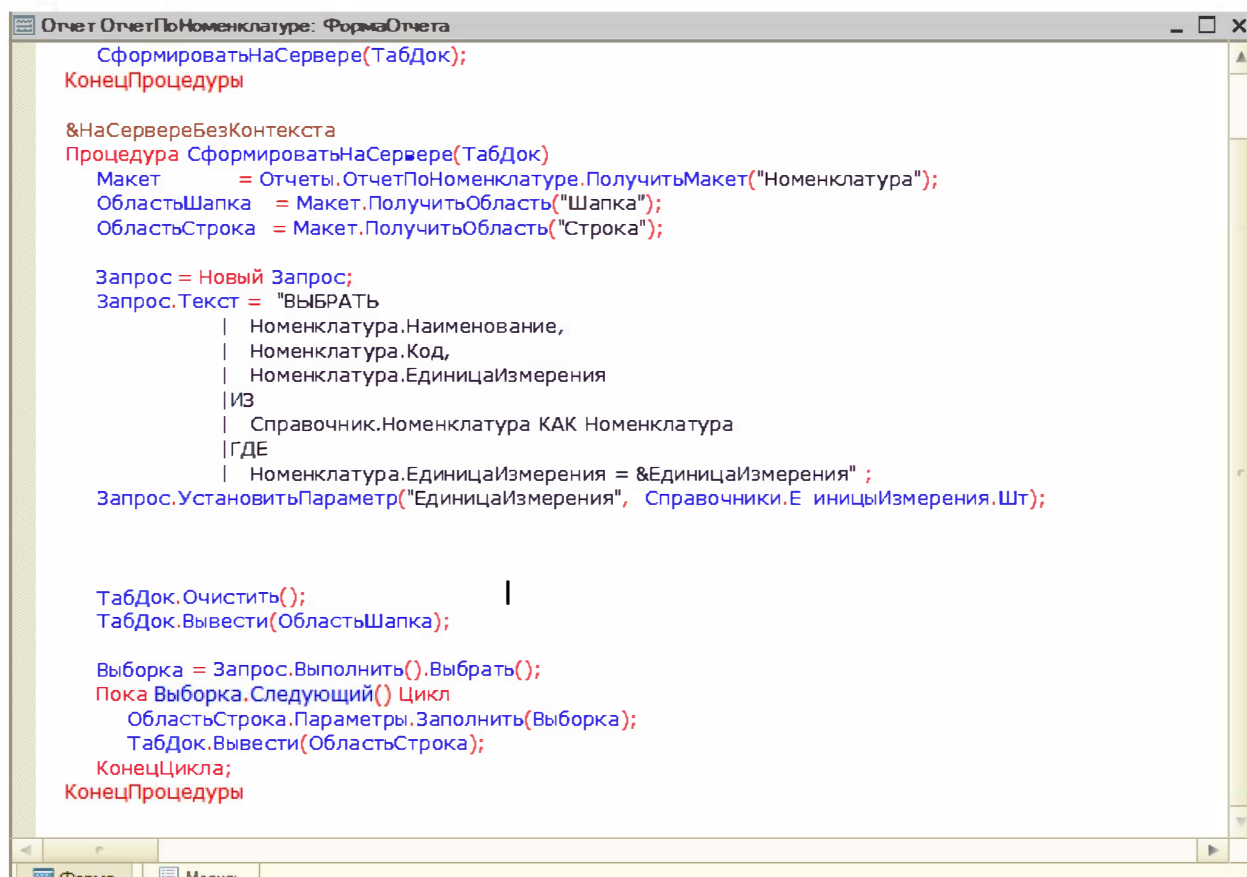
```
Запрос.Текст = "ВЫБРАТЬ
| Номенклатура.Наименование,
| Номенклатура.Код,
| Номенклатура.ЕдиницаИзмерения
|ИЗ
| Справочник.Номенклатура КАК Номенклатура
|ГДЕ
| Номенклатура.ЕдиницаИзмерения = &ЕдиницаИзмерения" ;
```

Рисунок 33

В нем появилось условие «ГДЕ Номенклатура.ЕдиницаИзмерения = &ЕдиницаИзмерения». Обратите внимание, как написан параметр: перед ним стоит символ «&» (амперсанд). Именно так выделяются параметры в тексте запроса, ведь сам текст запроса – это строка, которую можно написать руками. Нам пока этого делать не нужно, поэтому просто запомним, что параметры начинаются с амперсанда.

Настало время передать параметр в запрос. Как же нам это сделать? Для этого у объекта запрос существует метод «УстановитьПараметр(«ИмяПараметра», ЗначениеПараметра)», который в качестве параметров (простите за каламбур) принимает наименование параметра в виде строки и его значение. Чтобы установить параметр надо вызвать метод с параметрами. Не слишком ли сложно? Голова кругом. Не расстраивайтесь, спустя небольшое время вы даже не будете задумываться над этим, настолько это будет просто.

После того как мы описали текст запроса, нам необходимо так установить значение параметра для запроса. Итак, пишем (Рисунок 34).



```
СформироватьНаСервере(ТабДок);
КонецПроцедуры

&НаСервереБезКонтекста
Процедура СформироватьНаСервере(ТабДок)
    Макет = Отчеты.ОтчетПоНоменклатуре.ПолучитьМакет("Номенклатура");
    ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
    ОбластьСтрока = Макет.ПолучитьОбласть("Строка");

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | Номенклатура.Наименование,
        | Номенклатура.Код,
        | Номенклатура.ЕдиницаИзмерения
        | ИЗ
        | Справочник.Номенклатура КАК Номенклатура
        | ГДЕ
        | Номенклатура.ЕдиницаИзмерения = &ЕдиницаИзмерения ";
    Запрос.УстановитьПараметр("ЕдиницаИзмерения", Справочники.ЕдиницаИзмерения.Шт);

    ТабДок.Очистить();
    ТабДок.Вывести(ОбластьШапка);

    Выборка = Запрос.Выполнить().Выбрать();
    Пока Выборка.Следующий() Цикл
        ОбластьСтрока.Параметры.Заполнить(Выборка);
        ТабДок.Вывести(ОбластьСтрока);
    КонецЦикла;
КонецПроцедуры
```

Рисунок 34

Первым параметром метода «УстановитьПараметр» мы передаем строковое значение с именем параметра таким, каким оно задано в запросе, только здесь мы не используем амперсанд. А вот вторым параметром передаем само значение. Поскольку единица измерения «штука» является predefined элементом справочника (задана при

разработке конфигурации разработчиком прямо в конфигураторе), обратиться к этому элементу можно через точку из соответствующего справочника, что мы и сделали.

Посмотрим, как же изменился набор данных, получаемый из запроса. Сохраним изменения в конфигурации, нажав F7. Запустим конфигурацию в режиме «1С:Предприятие» и посмотрим на отчет (Рисунок 35).



Рисунок 35

Как видим, данных стало значительно меньше и у всех них единица измерения равна «Шт».

Однако условия могут быть заданы не только параметрами, также могут использоваться строковые или числовые литералы. Литералы – это некие фиксированные значения. Например, если я напишу: «Имя = 'Петя '»;», то "Петя " – это и есть литерал.

Зададим следующее условие: код должен равняться значению «000000032». Для этого откроем конструктор запроса, перейдем на вкладку «Условие» и установим флажок «Произвольное». Затем, отредактируем условие следующим образом (Рисунок 36).

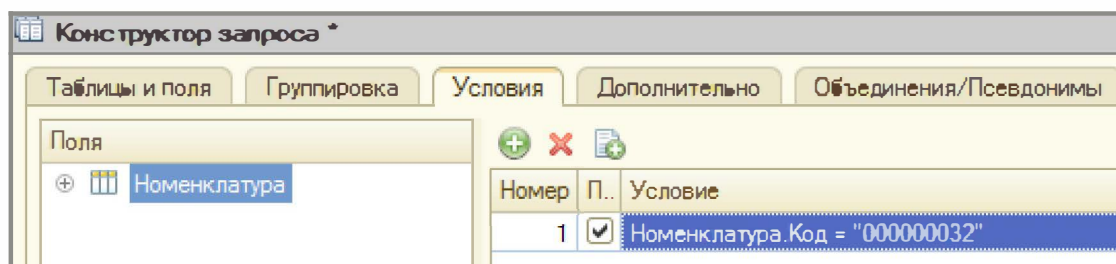


Рисунок 36

Сохраним изменения и снова сформируем отчет. Результатом будет отчет, содержащий лишь одну строку с той номенклатурой, которая удовлетворяет нашему условию.

1.6 Консоль запросов.

В этой части мы познакомимся с очень полезным и нужным инструментом разработчика – консолью запросов. Консоль запросов – это разработка фирмы 1С, которая предназначена для разработки и отладки запросов. Работает она в пользовательском режиме, поэтому в большинстве случаев конфигуратор и отладка не нужны для создания и отладки запросов. Скачать консоль запросов можно с сайта фирмы 1С по ссылке: <https://its.1c.ru/db/metod8dev#content:4500:hdoc>.

В нашу конфигурацию также встроена консоль запросов, она несколько отличается от консоли фирмы 1С, но работает примерно также. Для того чтобы ее вызвать в пользовательском режиме необходимо выбрать «Склад» -> «Сервис» -> «Консоль запросов». Рассмотрим ее интерфейс. Он состоит из нескольких разделов (Рисунок 37).

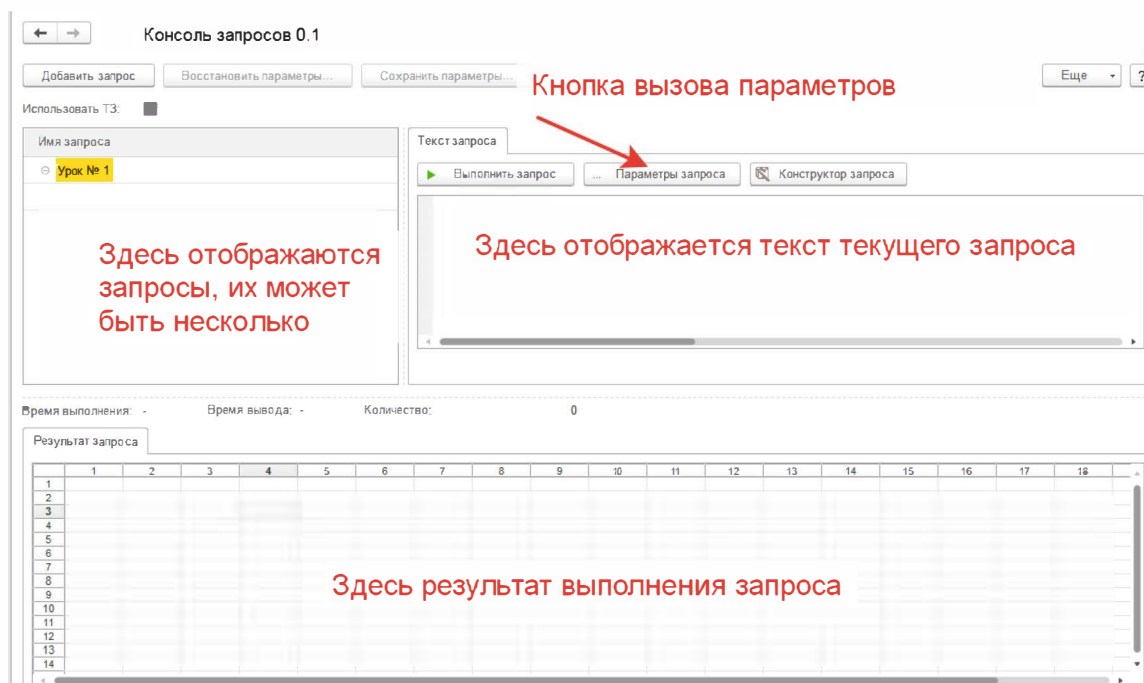


Рисунок 37

Интерфейс, как видим, интуитивно понятен и не должен вызывать каких-либо сложностей. Давайте попробуем создать запрос аналогичный тому, что мы делали в конфигураторе. Для начала в окне имен запросов (вверху слева) введем имя нашего запроса. Назовем его «Номенклатура» (Рисунок 38).

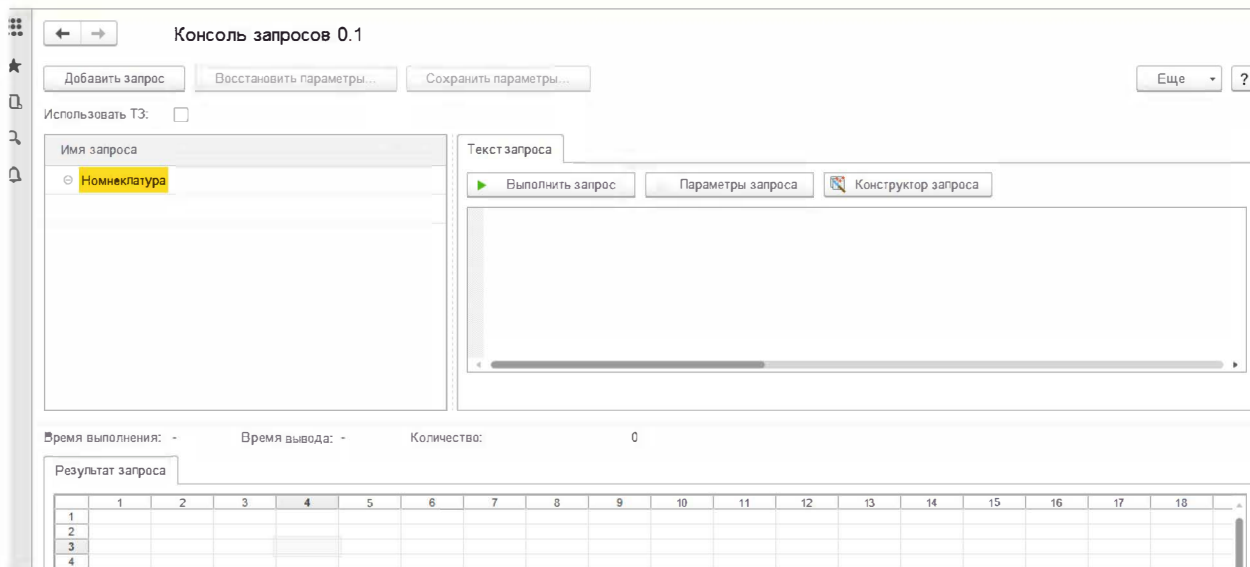


Рисунок 38

Нажимаем на кнопку «Конструктор запроса», формируем запрос к справочнику «Номенклатура», аналогично тому, что мы делали ранее. Работа с конструктором запроса ничем не отличается от того, что было в конфигураторе. Выбираем те же поля: «Наименование», «Код», «ЕдиницаИзмерения». Далее на закладке условия задаем условие по единице измерения (Рисунок 39-40).

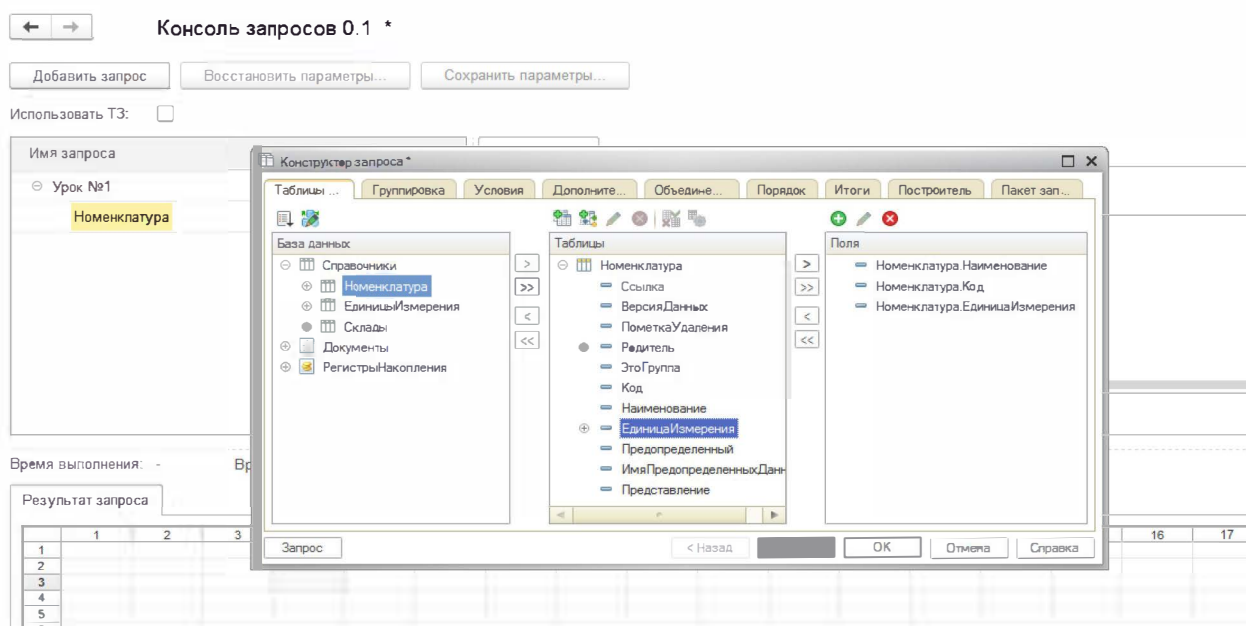


Рисунок 39

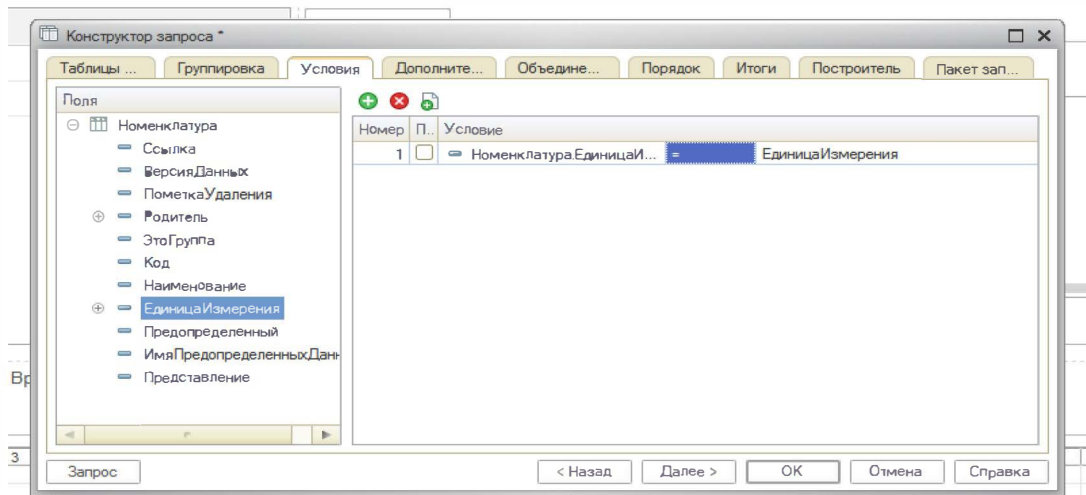


Рисунок 40

Нажимаем «Ок». Видим наш вновь сформированный текст запроса (Рисунок 41).

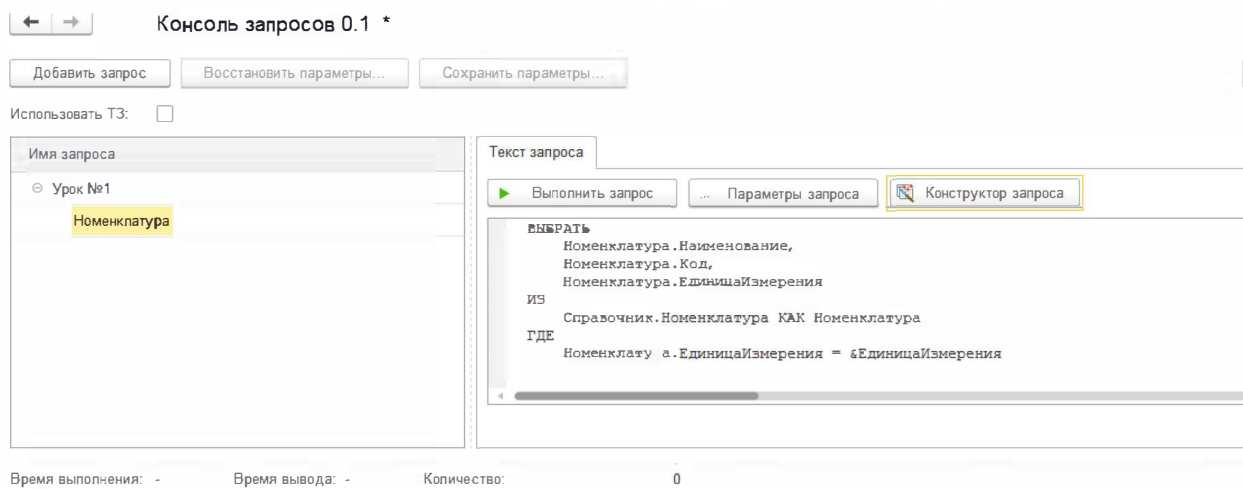


Рисунок 41

Теперь, для того чтобы установить параметры, нажимаем на кнопку «Параметры запроса». В открывшемся окне мы можем сами добавить требуемый параметр, но поскольку мы только начинаем осваивать консоль запросов и нам может показаться это сложновато, то проще нажать на кнопку «Обновить параметры» и параметры из запроса будут автоматически сюда добавлены. Дважды щелкаем по полю «Значение» и выбираем «Показать все», откроется форма выбора единиц измерения, выбираем «Шт» (Рисунок 42).

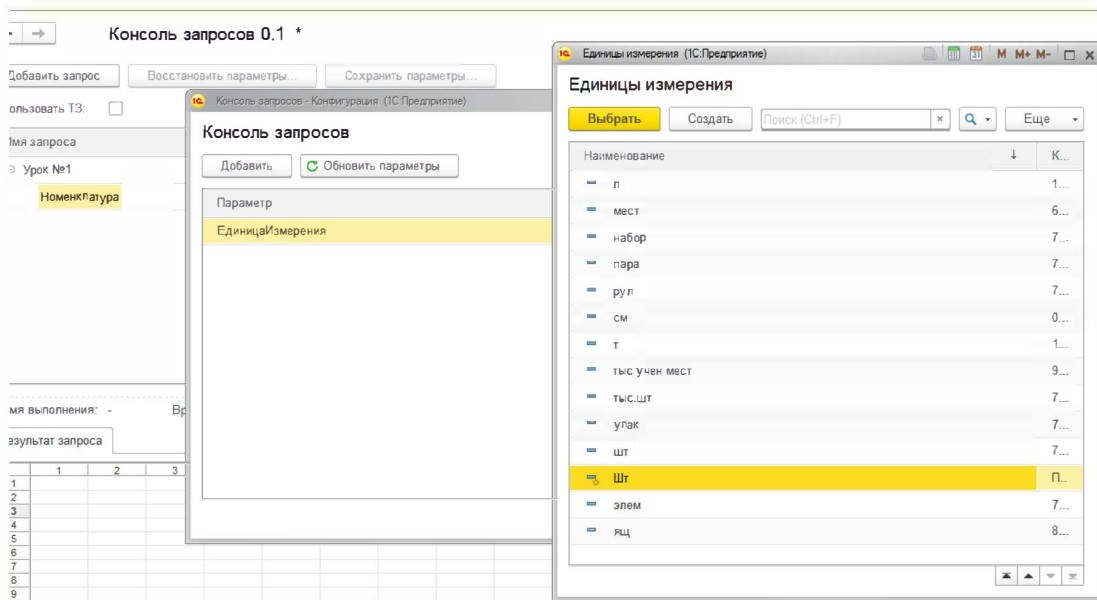


Рисунок 42

Нажимаем «Выбрать» и «Ок». Теперь параметр установлен. Нажимаем на кнопку «Выполнить запрос» и видим результат выполнения нашего запроса (Рисунок 43).

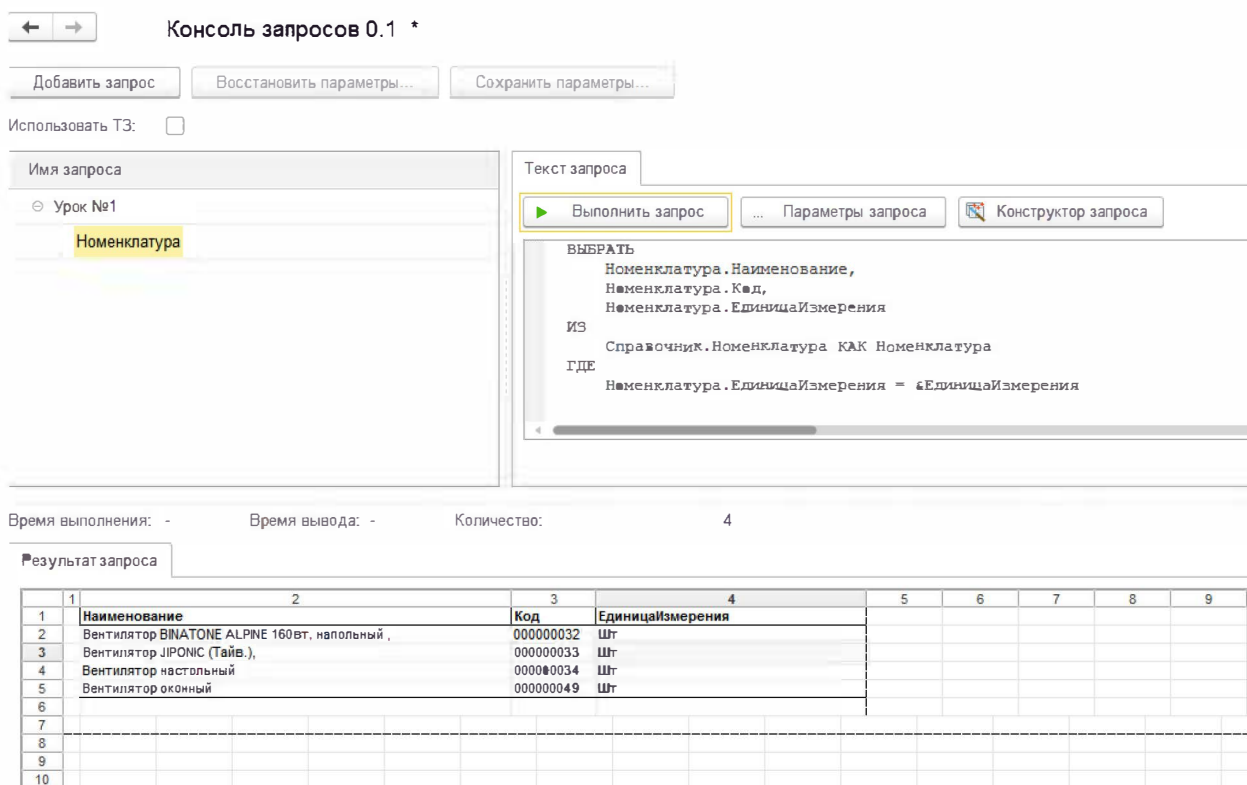


Рисунок 43

Таким образом, в пользовательском режиме исключительно декларативно (не написав ни строчки кода) мы разработали запрос, выполнили его и увидели результат выполнения.

Как видите, уважаемые читатели, консоль запросов является весьма удобным и полезным инструментом. Одним из приемов разработки является разработка и отладка запроса в консоли, а затем его перенос в программный код.

1.7 Псевдонимы полей.

Ранее при разработке запросов мы не задумывались над тем, как называются наши поля в результате, но если мы посмотрим, то увидим, что называются они так же, как поля нашего справочника. Но это не всегда бывает удобно. Например, нам бы хотелось, чтобы поле «ЕдиницаИзмерения» называлось «ЕдИзм». Как это сделать? Нет ничего проще. Для этого открываем конструктор запроса и переходим на закладку «Объединения/Псевдонимы». Там мы увидим все выходные поля результата и их имена. Дважды щелкнем левой кнопкой мыши по полю «ЕдиницаИзмерения» и отредактируем его как обычную строку (Рисунок 44).

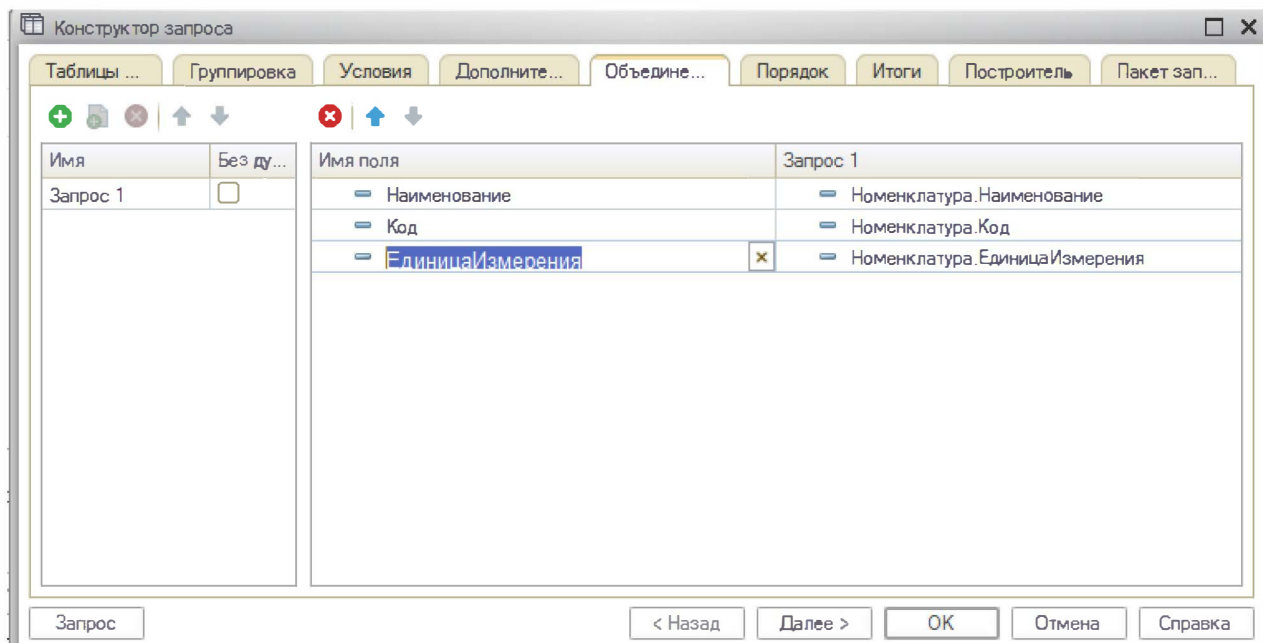


Рисунок 44

Посмотрим, как изменился текст запроса. У поля «ЕдиницаИзмерения» добавилось ключевое слово «КАК», после которого дописано новое имя поля «ЕдИзм» (Рисунок 45).

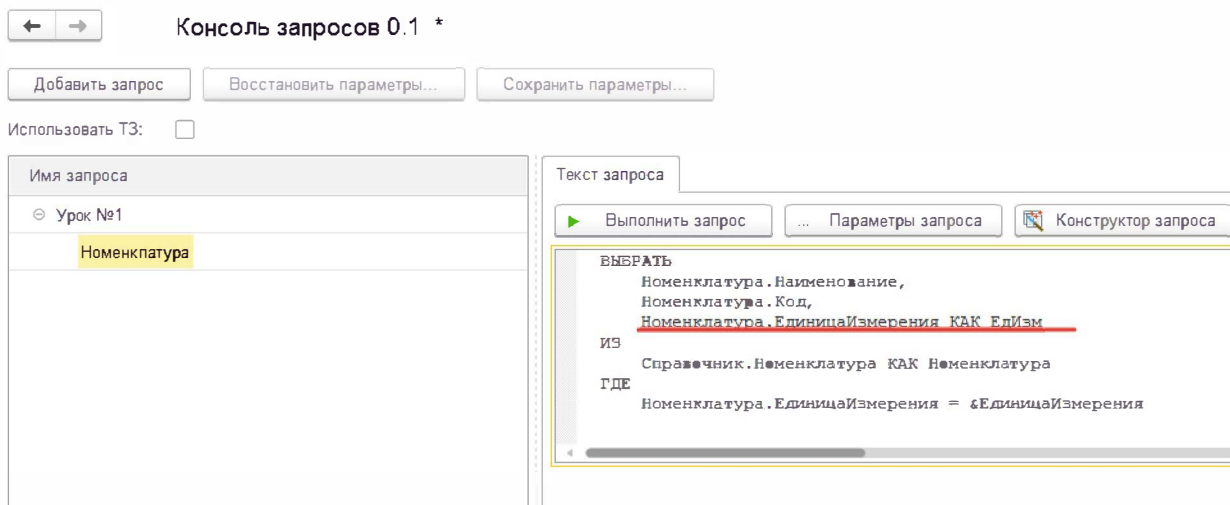


Рисунок 45

Если мы изменяем имена полей (другими словами, устанавливаем им псевдонимы), то и в дальнейшем в программном коде мы должны обращаться к этим полям по новым именам (Рисунок 46).

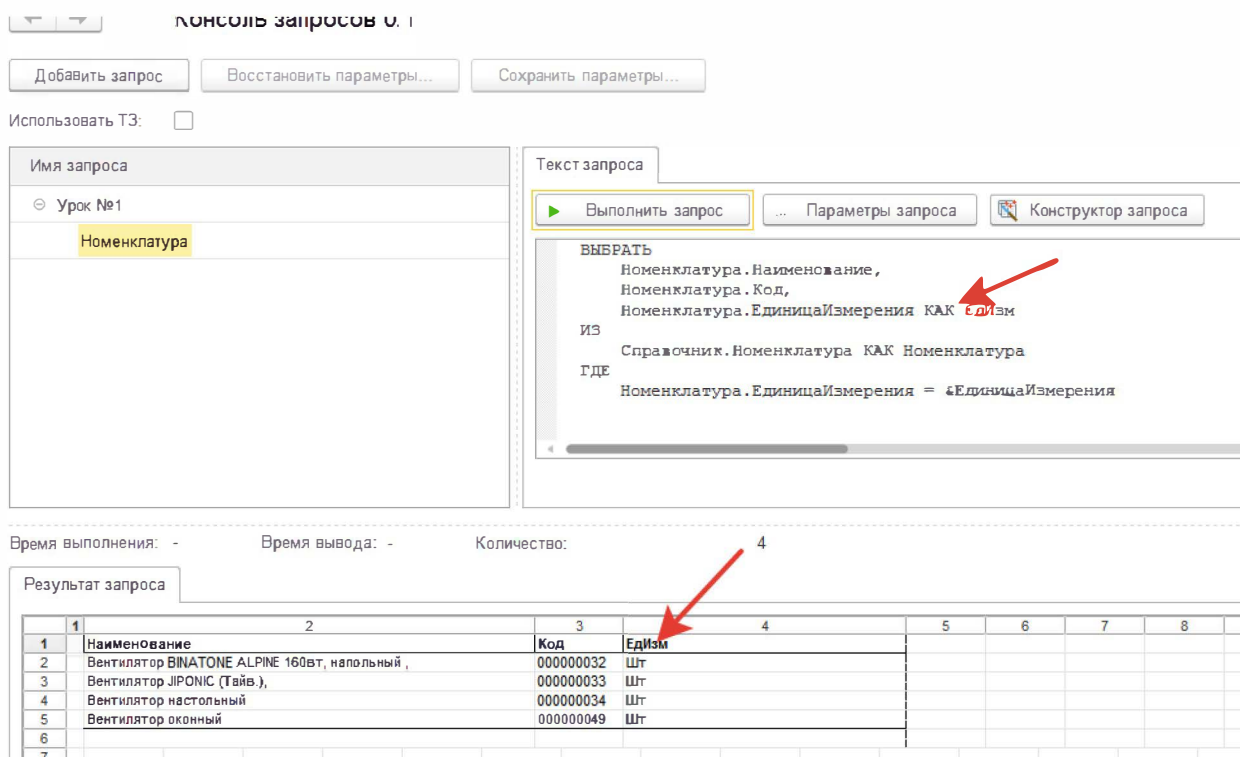


Рисунок 46

1.8 Сортировка.

То, в каком порядке расположены данные в результате выполнения запроса, в некоторых случаях может иметь значение. Именно поэтому важно уметь сортировать полученный результат. Это делается в конструкторе запроса на закладке «Порядок». На этой закладке мы выбираем поля, по которым хотим выполнять сортировку и направление сортировки: возрастание, убывание, иерархия, иерархия по убыванию. Если полей будет выбрано несколько, то сортировка будет выполнена сначала по первому полю, затем, если есть одинаковые значения по первому полю, то будет выполнена сортировка внутри этих одинаковых значений по второму полю и так далее.

Например, мы имеем выборку из трех полей: «фамилия», «имя», «отчество» и задаем по ним сортировку. Сначала сортировка выполнится по полю «фамилия», но представьте, что с одинаковой фамилией есть несколько человек. Эти несколько человек расположатся в порядке их имен. Ну а если среди элементов есть элементы с одинаковой фамилией и именем, то они будут отсортированы между собой по отчеству. В нашем примере отсортируем результат по полю «Код». Выберем его на закладке «Порядок» (Рисунок 47).

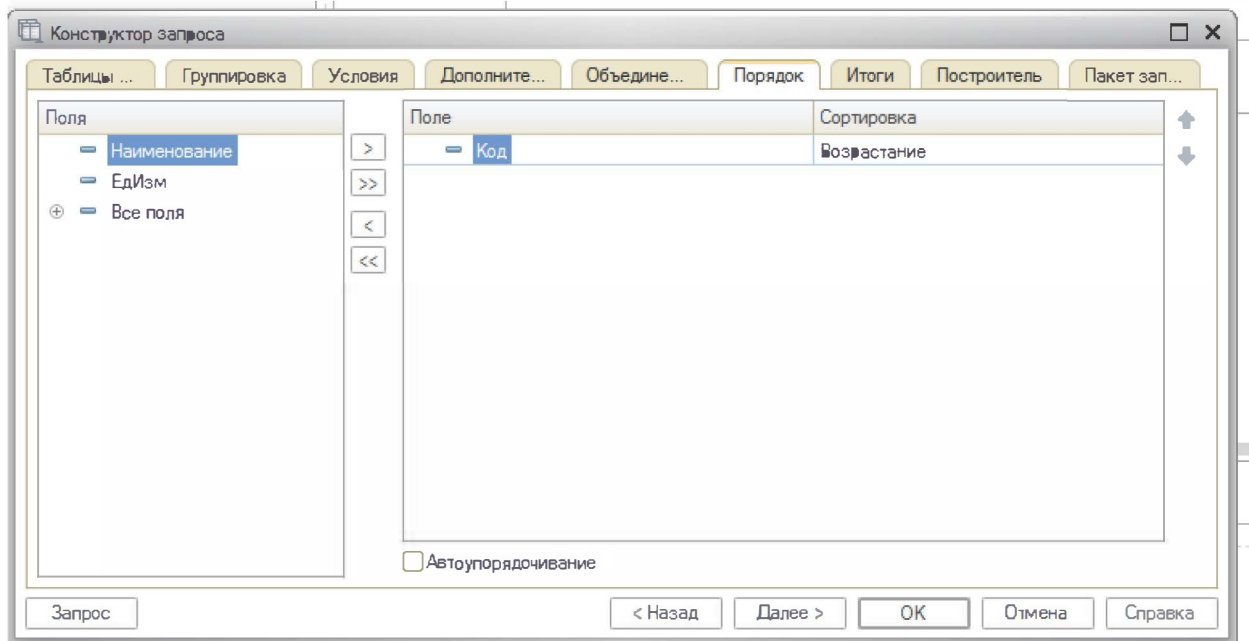


Рисунок 47

Как изменился текст запроса? Мы видим новое ключевое слово «Упорядочить по», а затем имя поля упорядочивания (Рисунок 48).

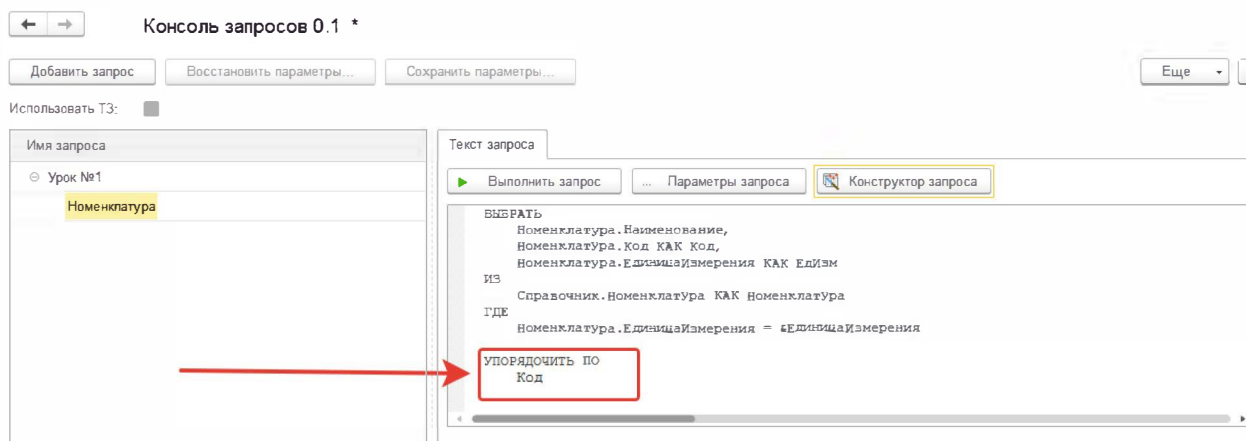


Рисунок 48

Слово «Возр» опущено, это допускается, если упорядочивание идет по возрастанию. Если же мы выберем порядок по убыванию, то увидим слово «УБЫВ» рядом с полем упорядочивания.

Теперь попробуем выполнить сортировку по нескольким полям. Для этого сначала уберем условие, чтобы в выборку попали все элементы, затем первым полем сортировки назначим единицу измерения по возрастанию, а вторым – код по убыванию. Выполним запрос (Рисунок 49-50).

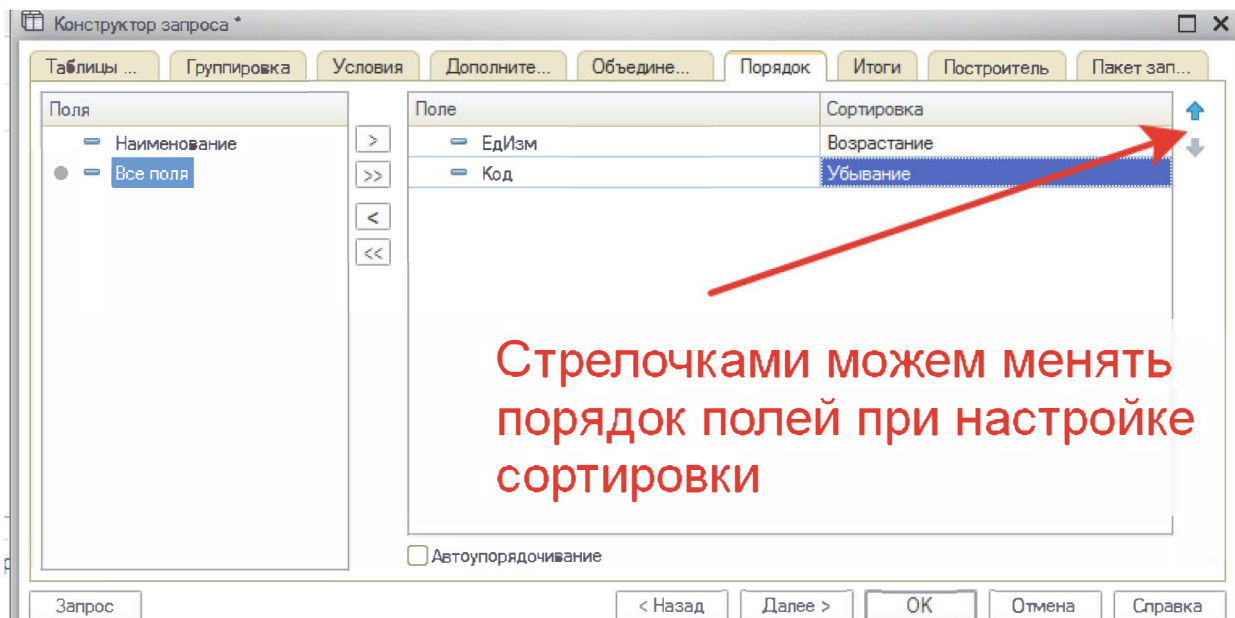


Рисунок 49

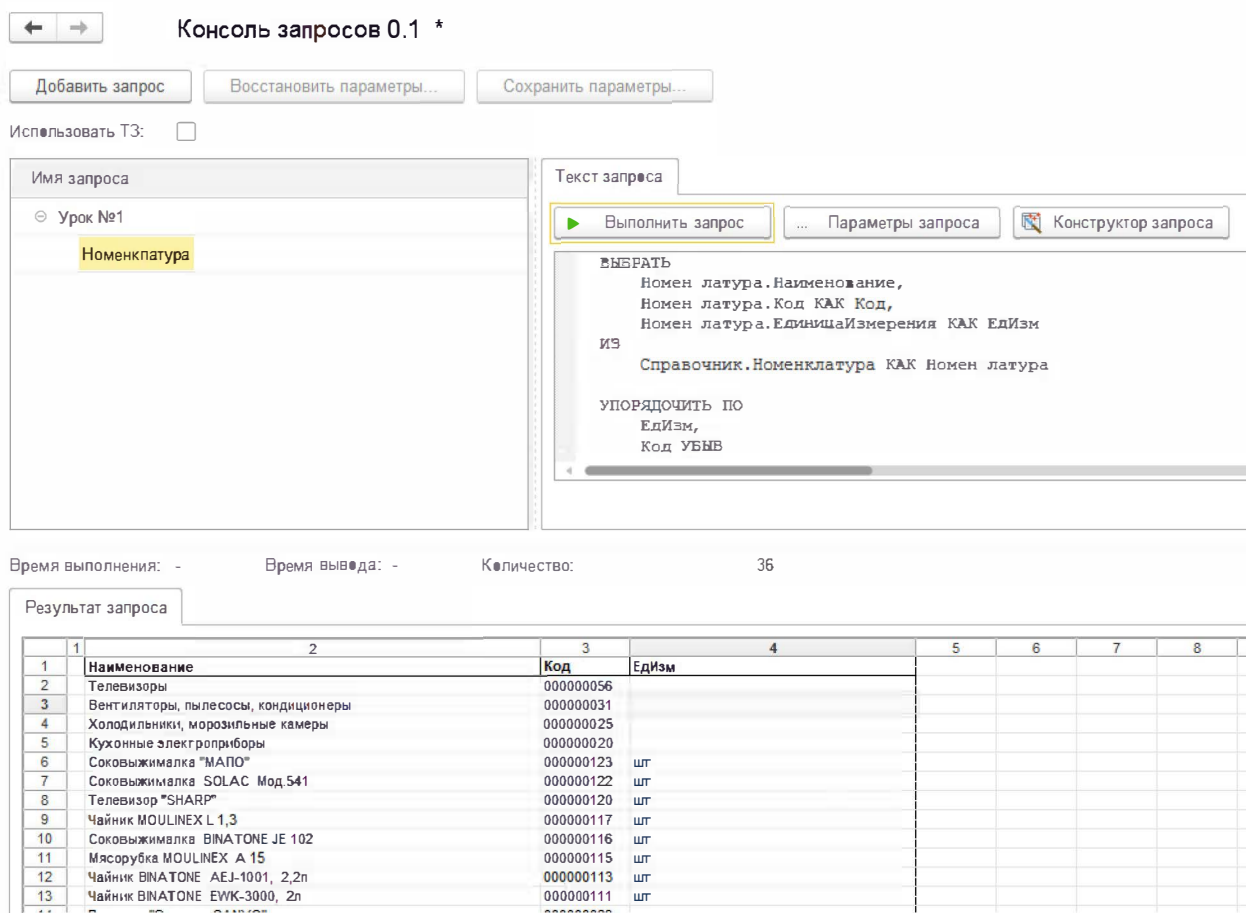


Рисунок 50

В результате мы видим, что данные отсортированы по единице измерения, а внутри этой сортировки – по коду в порядке убывания. То есть те элементы, у которых одинаковая единица измерения, между собой отсортированы по коду в порядке убывания.

Сортировка может быть выполнена по любому полю выбранной таблицы, даже если это поле не было выбрано в запросе. Мы можем отсортировать наши данные по полю

«ЭтоГруппа», хотя такое поле мы не выбирали в результат. «ЭтоГруппа» – реквизит с типом данных булево (принимает значения «Истина» или «Ложь»), который является признаком группы.

1.9 Получить первые.

В этой части речь пойдет о том, как получить первые несколько элементов из всей выборки.

Перед нами стоит задача получить первые 5 элементов справочника «Номенклатура». Для этого открываем конструктор запросов, удаляем все условия, если они есть, переходим на закладку «Дополнительно». Видим там группу «Выборка записей», устанавливаем флажок «Первые», он становится активным после ввода количества первых записей, устанавливаем в нем значение 5 (Рисунок 51).

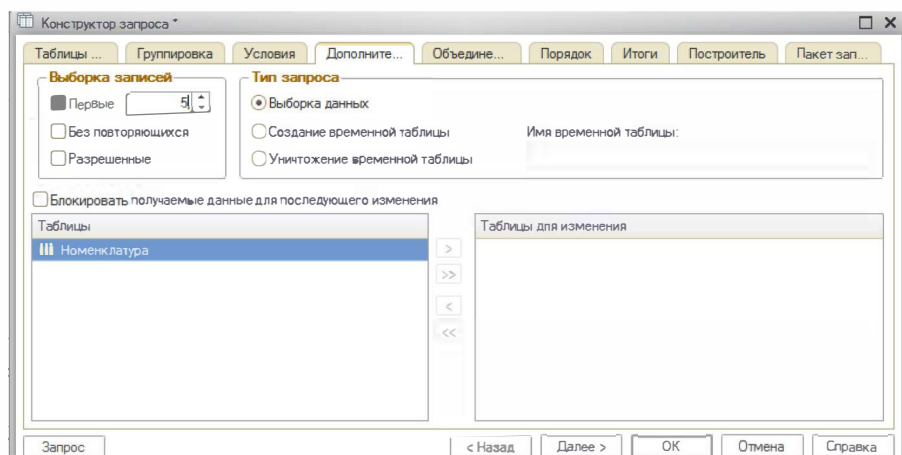


Рисунок 51

Нажимаем «Ок» и традиционно смотрим, как изменился текст нашего запроса (Рисунок 52).

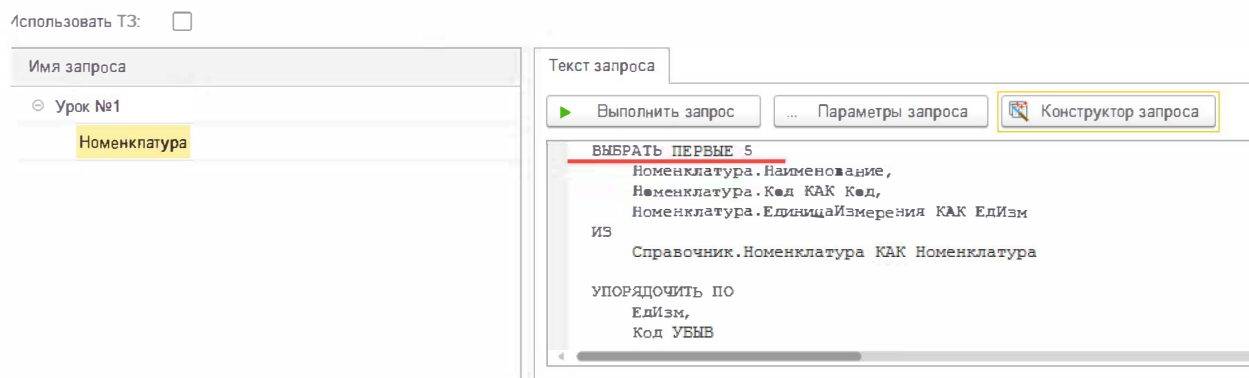


Рисунок 52

Появилось ключевое слово «Первые 5» после слова «Выбрать». А вот тут необходимо обратить внимание на то, что сортировка играет ключевую роль в том, какие записи попадут в эти первые 5. Если мы хотим получить элемент с самым большим кодом, то нам
31 Верный старт в 1С www.work-1c.ru 88002344322 Бесплатный звонок

необходимо установить сортировку по коду в порядке убывания и выбрать первые 1 элементов.

1.10 Без повторяющихся.

Для того чтобы продемонстрировать следующий пример, нам придётся изменить наш запрос. Уберем поля «Код» и «Наименование» и оставим только «ЕдиницаИзмерения». В результате получим много одинаковых записей (Рисунок 53-54).

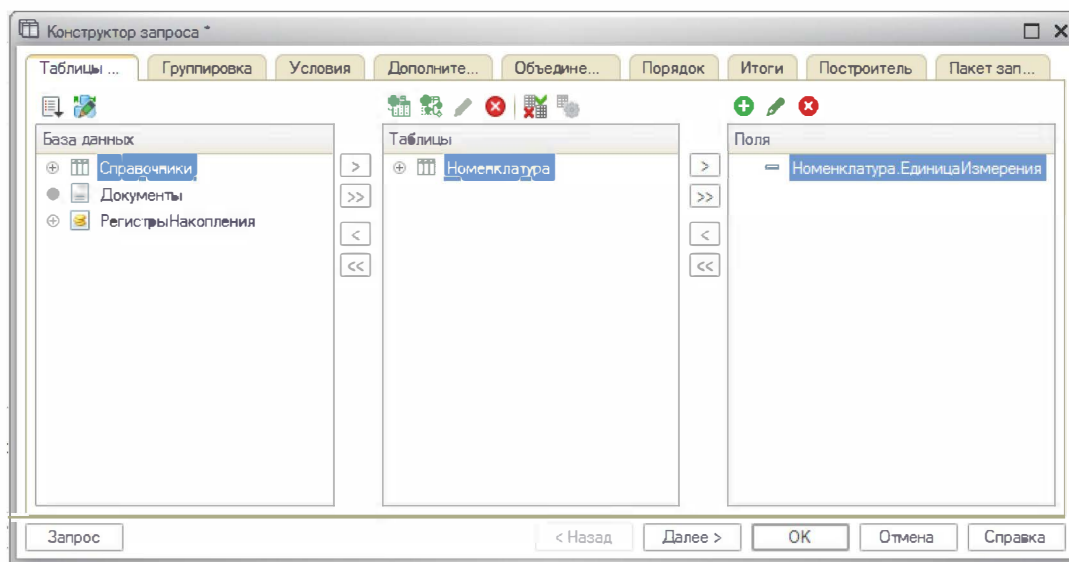


Рисунок 53

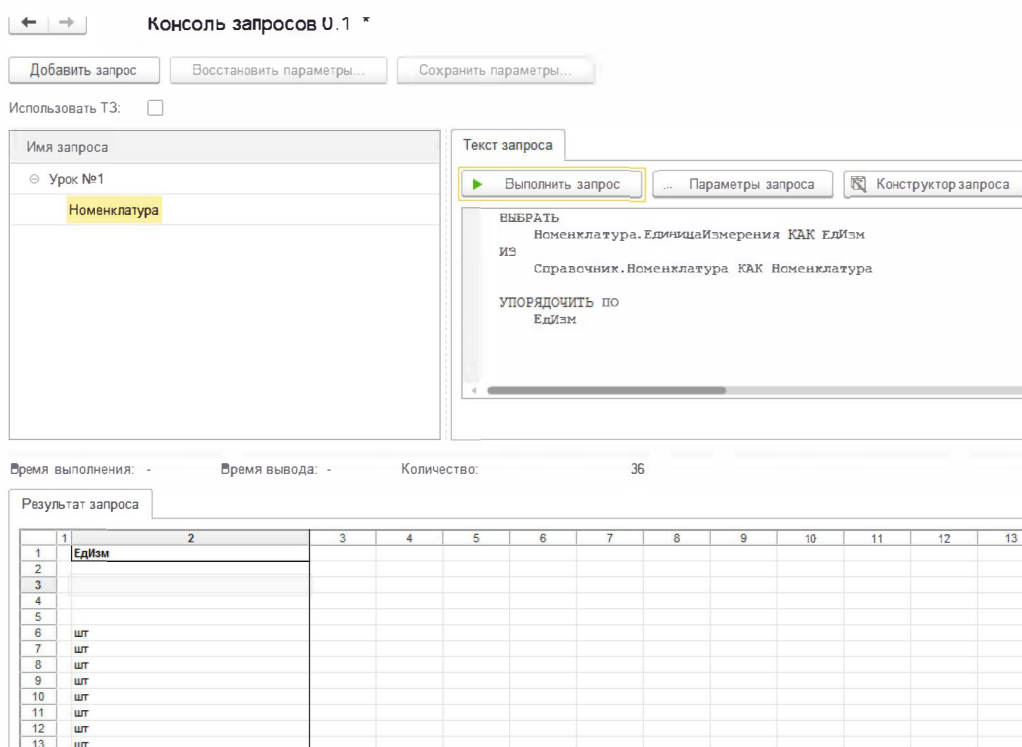


Рисунок 54

То есть, у нас есть повторяющиеся записи. Если нам понадобится вывести не повторяющиеся результаты, то мы можем воспользоваться специальной возможностью в конструкторе запроса на закладке «Дополнительно».

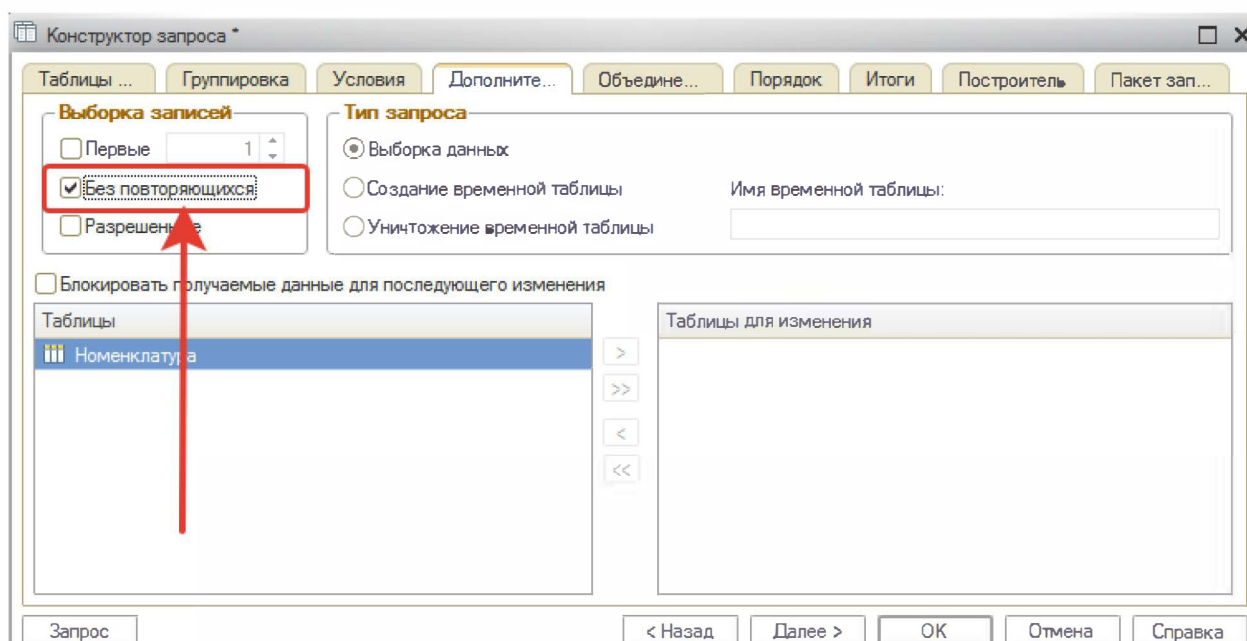


Рисунок 55

Посмотрим, как изменился текст запроса (Рисунок 56).

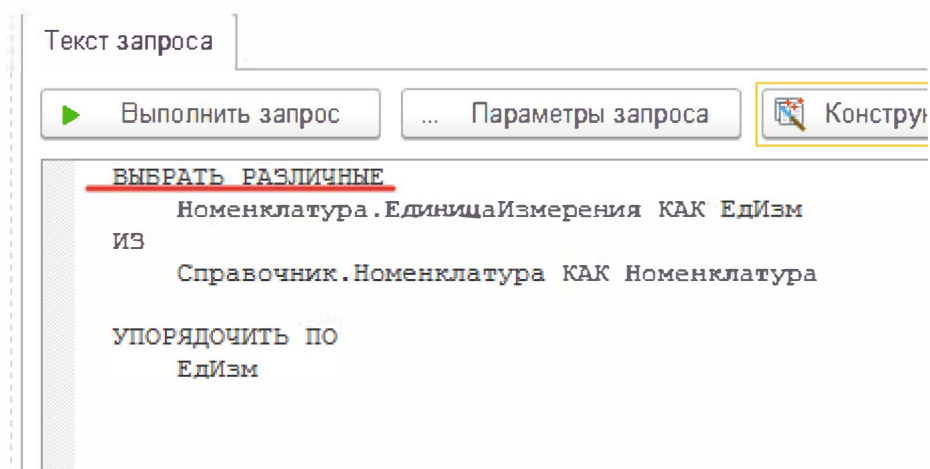


Рисунок 56

Появилось ключевое слово «Различные» после «Выбрать». Смотрим на результат, предварительно нажав на «Выполнить запрос» (Рисунок 57).

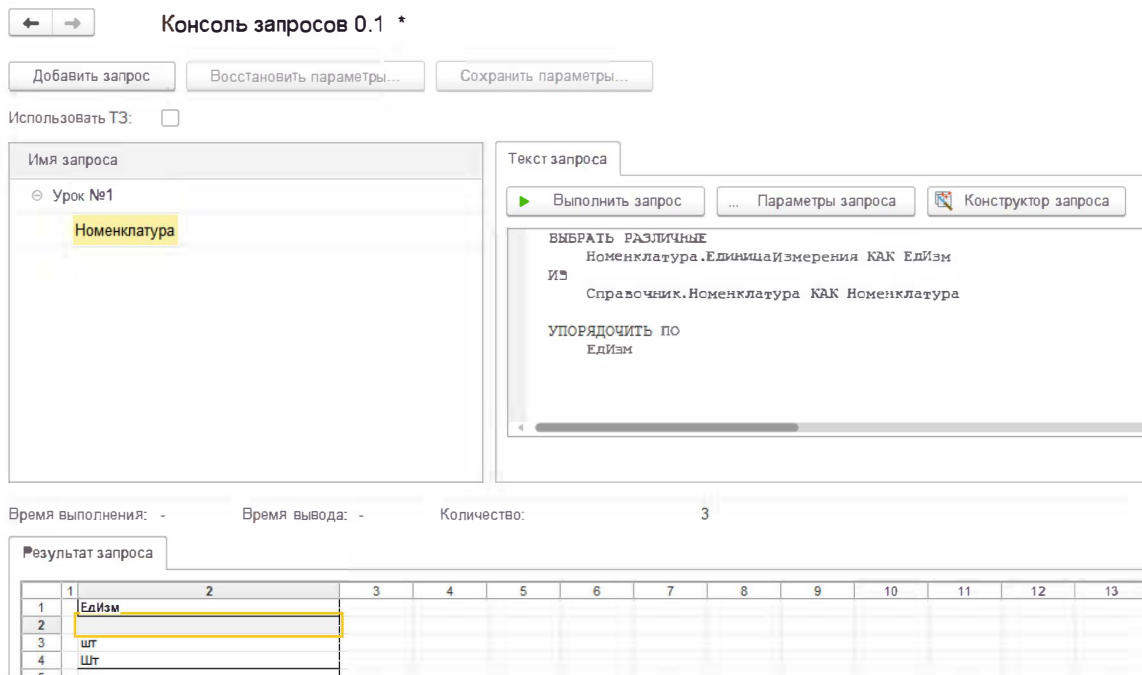


Рисунок 57

Мы видим только 3 строки: с незаполненной единицей измерения, с единицей «Шт» и единицей «шт». «Шт» и «шт», в данном случае, являются разными элементами справочника.

1.11 Реальные таблицы. Партии товаров.

В этом примере мы выполним запрос к регистру накопления «Партии товаров». Откроем конструктор запроса и выберем соответствующий регистр (Рисунок 58).

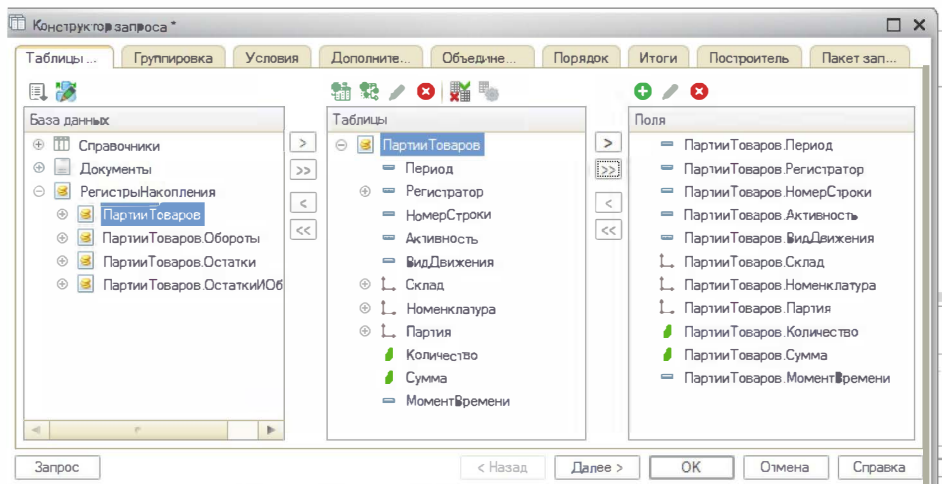


Рисунок 58

Эта таблица, «ПартииТоваров», физически существует в базе данных, поэтому мы называем ее реальной, три другие виртуальные, но о них позже.

Выполним запрос и посмотрим на результат. Мы видим все записи, которые были сделаны в ней документами за весь период (Рисунок 59).

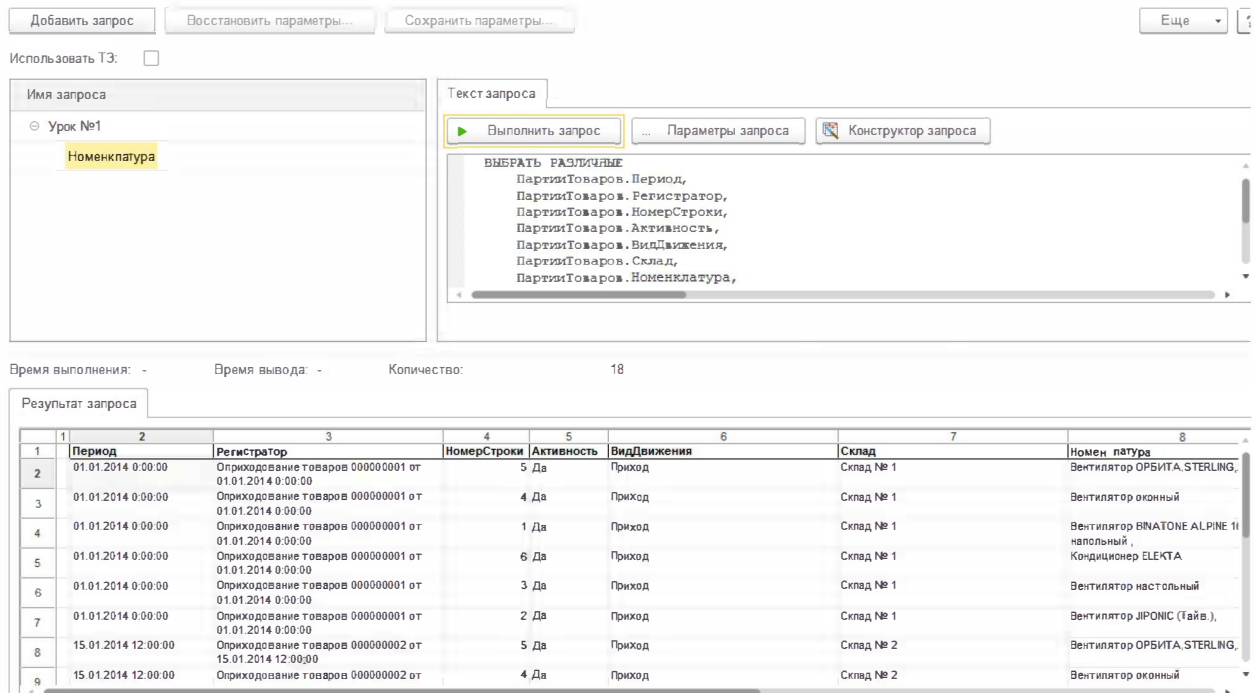


Рисунок 59

1.12 Условия. Между, сравнения.

Получим из регистра «ПартииТоваров» все записи с количество больше 100. Для этого в конструкторе переходим на вкладку «Условия» и устанавливаем его при помощи числового литерала (Рисунок 60).

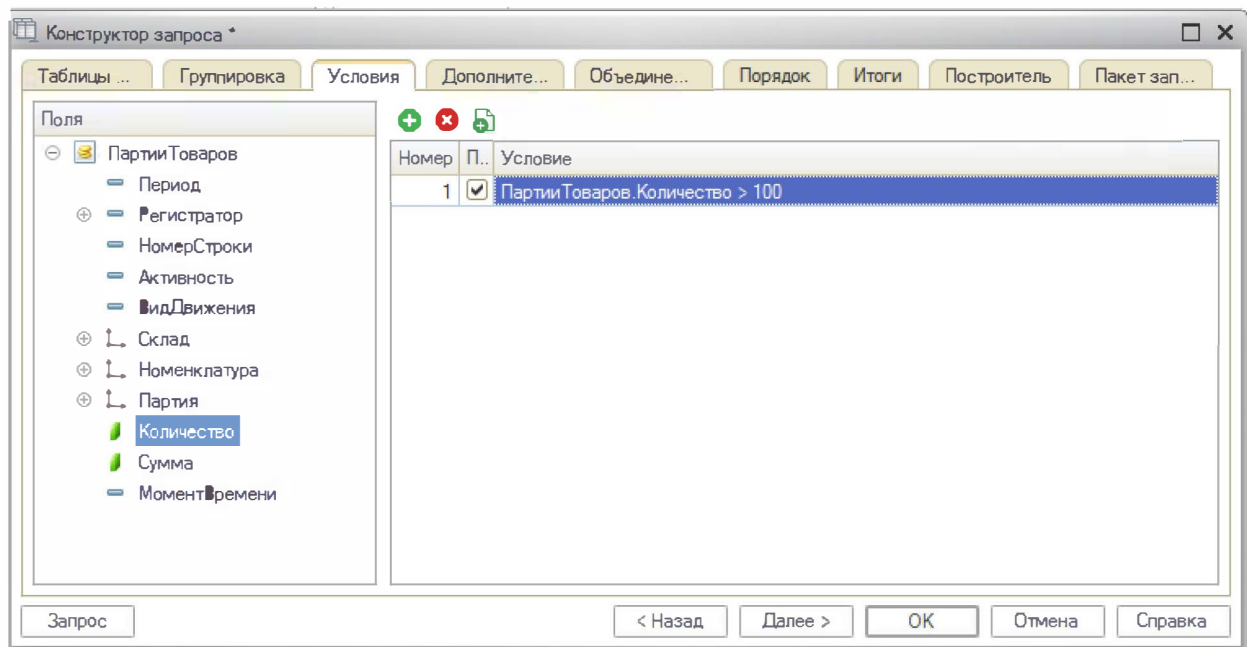


Рисунок 60

Выполняем запрос и проверяем результат (Рисунок 61).

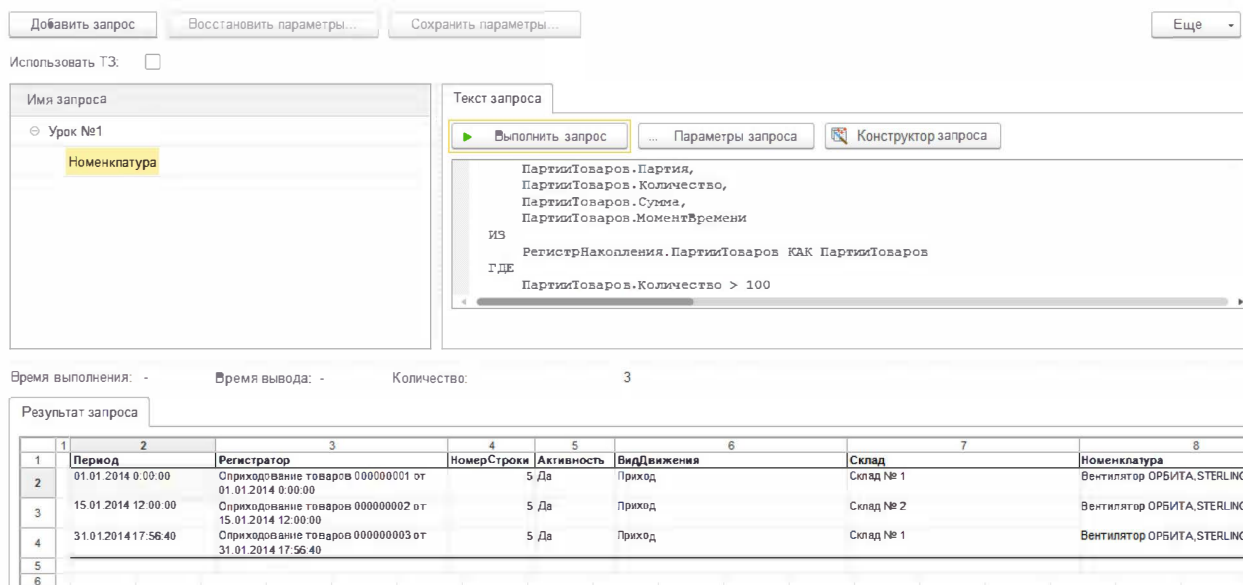


Рисунок 61

Видим, что у всех строк количество больше 100.

Попробуем использовать условие «Между», оно работает как $Условие1 \leq Значение \leq Условие2$. То есть то значение, на которое мы накладываем условие, больше либо равно правому значению и меньше либо равно левому. Наложим такое условие, опять же, при помощи числовых литералов. Пусть в нашу выборку попадут строки с количеством между 150 и 200 (Рисунок 62-63).

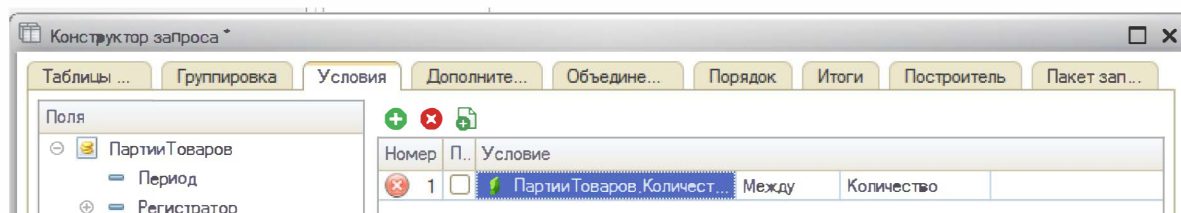


Рисунок 62

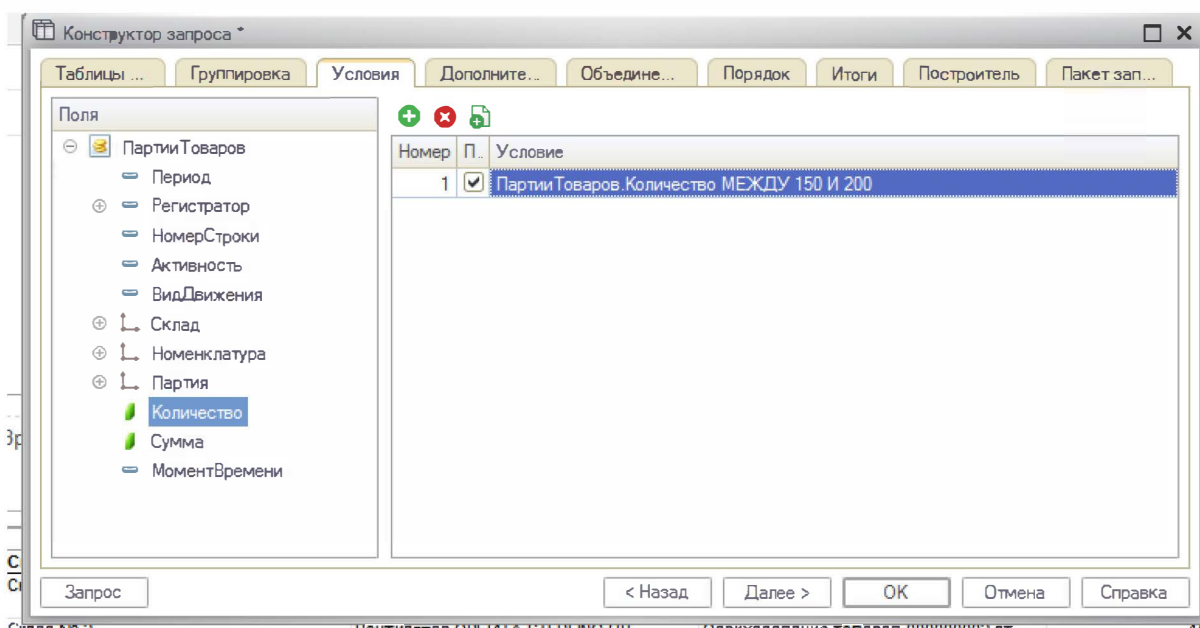


Рисунок 63

Для начала нам пришлось наложить простое условие с оператором «Между» и параметрами. Затем мы установили флажок «Произвольное» и заменили параметры литералами.

Мы могли добиться того же результата и другим путем, наложив 2 условия на поле «Количество», чтобы количество было ≥ 150 И ≤ 200 (Рисунок 64).

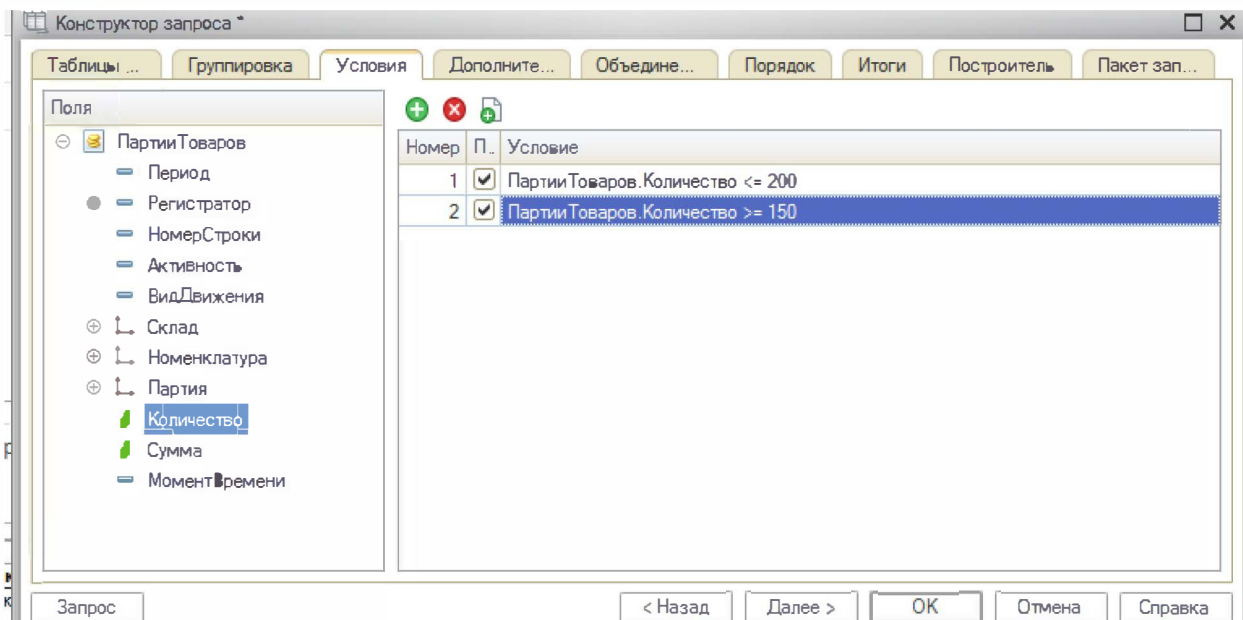


Рисунок 64

Результат будет таким же, но условие в первом случае получается компактнее (Рисунок 65-66).

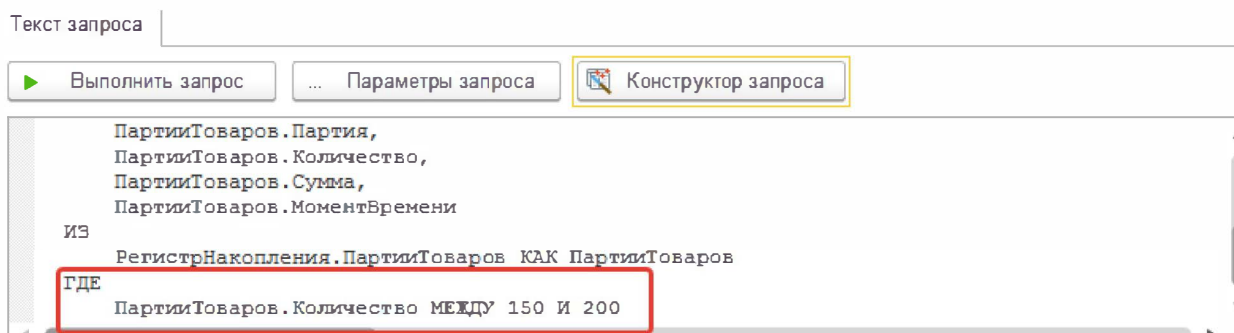


Рисунок 65

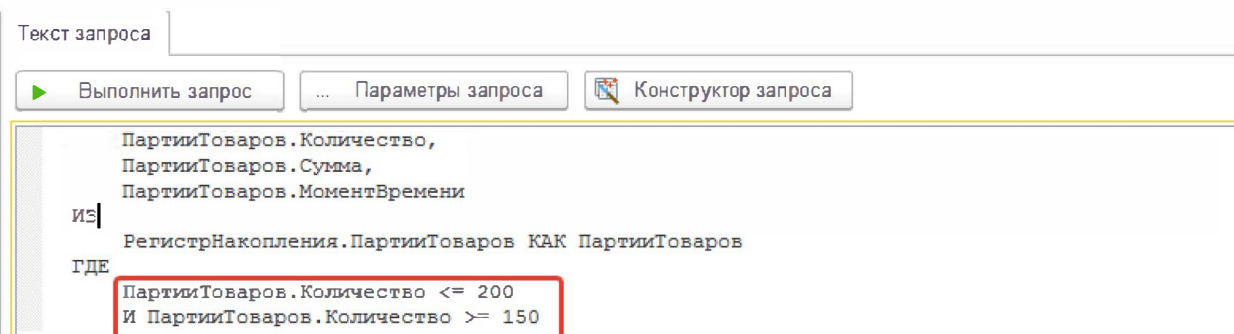


Рисунок 66

Схожим образом это работает для дат.

Допустим, нам нужны записи с датой раньше или равно 16 января. Дату зададим параметром (Рисунок 67).

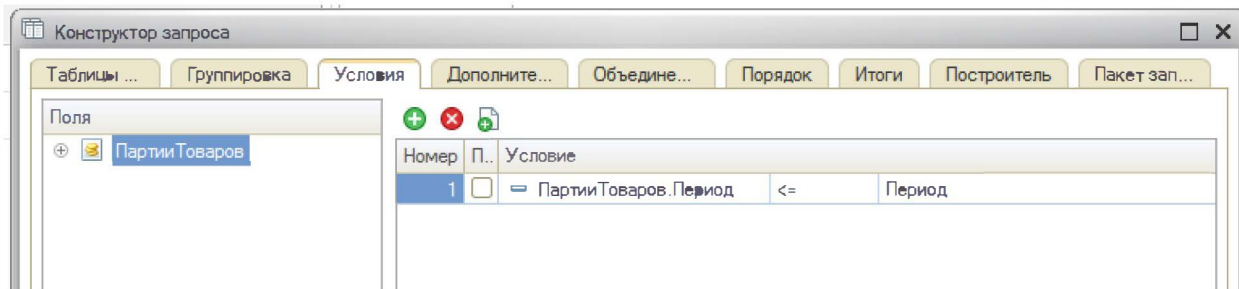


Рисунок 67

Нажимаем «ОК» и заполняем параметры (Рисунок 68).



Рисунок 68

Проверяем результат. В него попали записи от 1 и 15 января, что соответствует нашему условию (Рисунок 69).

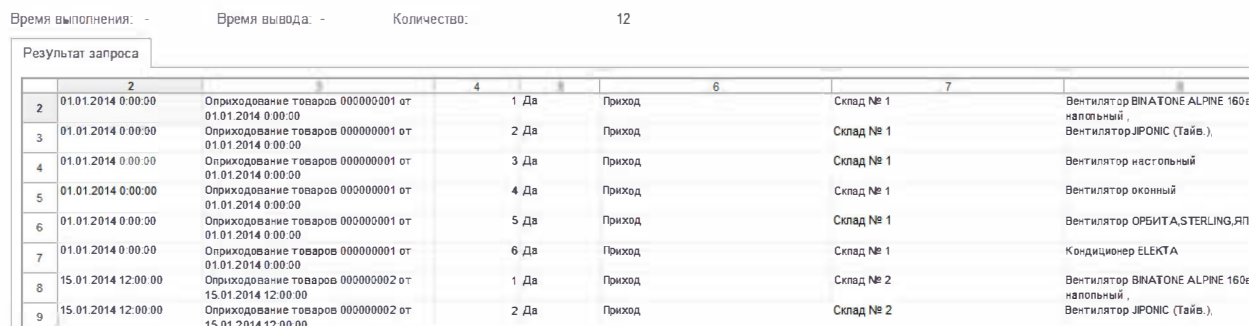


Рисунок 69

1.13 Виртуальные таблицы. Остатки.

Ранее мы упоминали о неких виртуальных таблицах. Здесь мы остановимся на них подробнее. Вкратце скажу, что виртуальные таблицы не существуют в базе данных физически, они создаются системой (платформой 1С) динамически для оптимизации работы и ускорения получения результатов. Обратимся к виртуальной таблице «Остатки». Она называется «ПартииТоваровОстатки». Как видим, в плане содержимого таблицы, постфикс является очень информативным. Для виртуальных таблиц при их добавлении в запрос становится активна кнопка «Параметры виртуальной таблицы» (Рисунок 70).

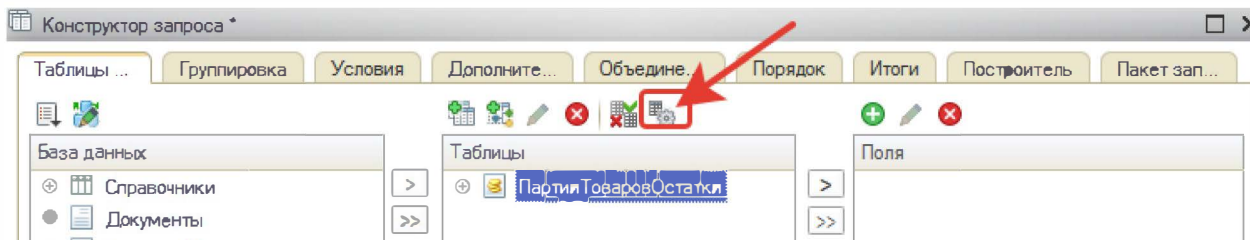


Рисунок 70

Здесь мы можем задать некие условия, по которым будет сформирована виртуальная таблица. Накладывать условия на закладке «Условия» для виртуальных таблиц является, как минимум, неоптимальным, а, как максимум, ошибочным. Какие же условия мы можем наложить в этих параметрах? Прежде всего, это «Период». Период задается параметром и обозначает дату, на которую требуется получить остатки.

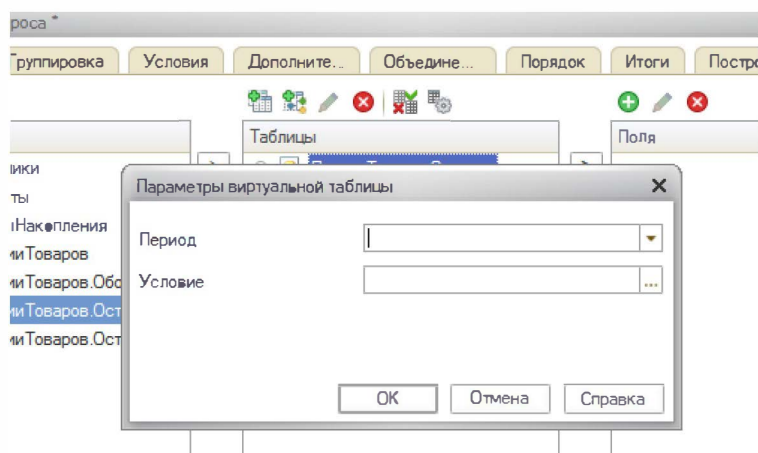


Рисунок 71

Еще есть поле «Условие». Чтобы отредактировать условие, необходимо нажать на кнопку выбора (Рисунок 72)

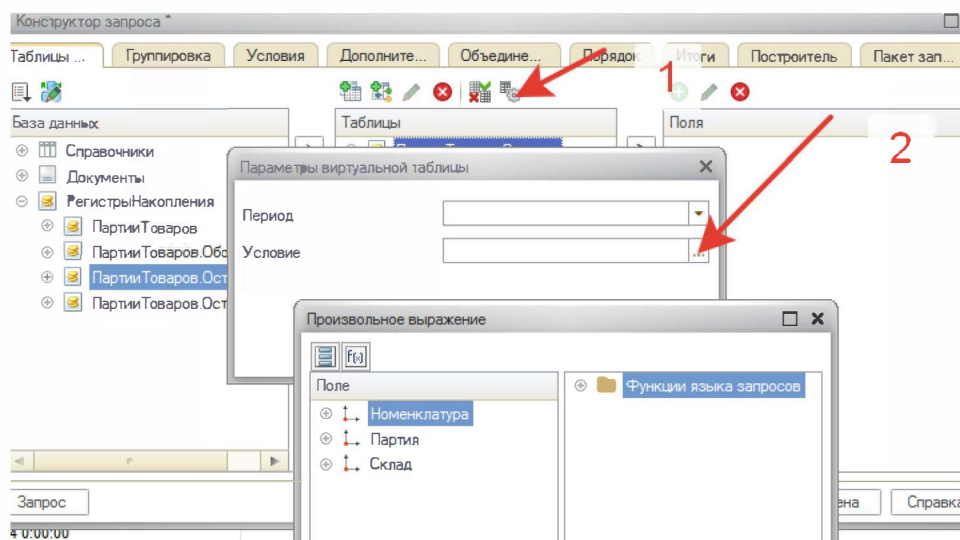


Рисунок 72

В открывшемся окне мы можем наложить произвольные условия на измерения регистра. Измерения регистра накопления представляют собой разрезы учета, по которым он ведется. В нашем случае их 3: «Склад», «Партия», «Номенклатура». Это значит, что мы

знаем не просто общее число товаров, но и количество каждого товара (разрез учета – номенклатура), не просто количество каждого товара, но и количество каждого товара на каждом складе (разрез учета – склад), не просто количество товаров по каждому товару и складу, но и количество каждого товара в партии (разрез учета – партия).

Установим дату получения остатков (Рисунок 73-76).

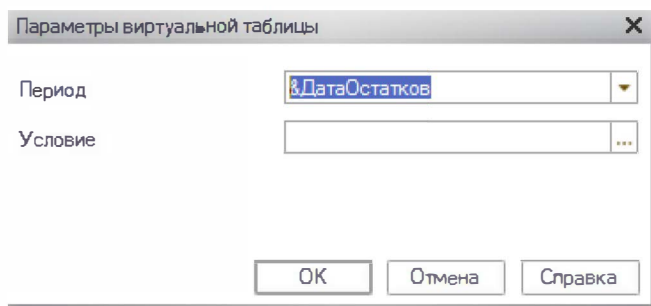


Рисунок 73

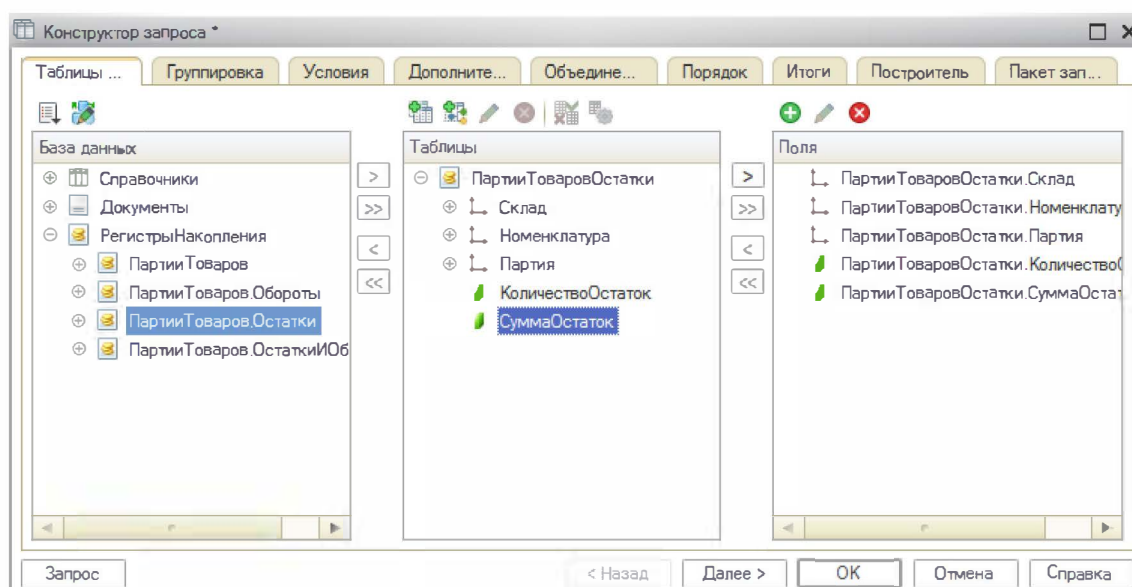


Рисунок 74

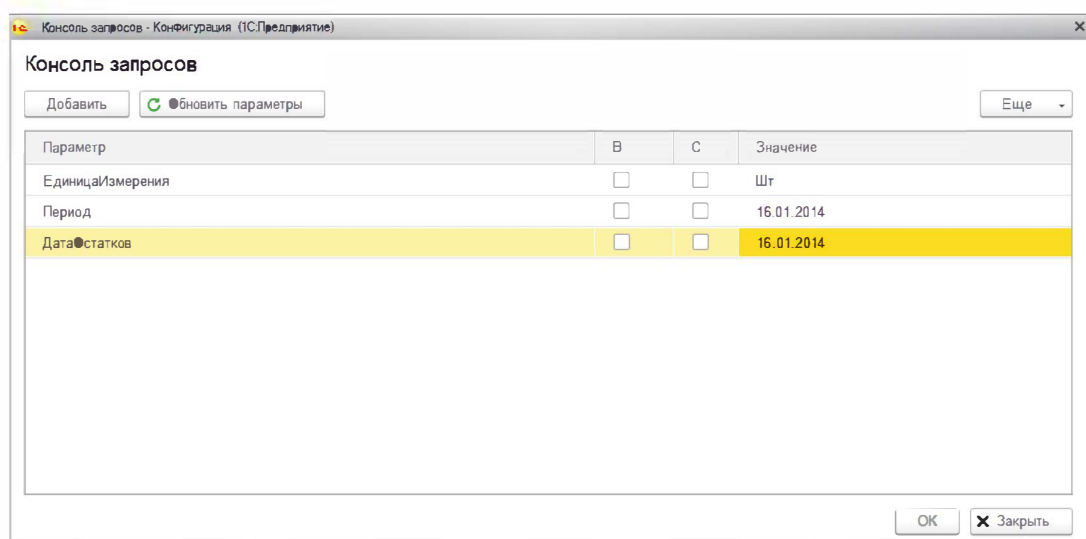


Рисунок 75

Время выполнения: - Время вывода: - Количество: 12

Результат запроса

1	2	3	4	5	6
7	Склад № 1	Кондиционер ELEKTA	Оприходование товаров 000000001 от 01.01.2014 0:00:00		44
8	Склад № 2	Вентилятор BINATONE ALPINE 160вт, напольный,	Оприходование товаров 000000002 от 15.01.2014 12:00:00		90
9	Склад № 2	Вентилятор J'RONIC (Тайв.),	Оприходование товаров 000000002 от 15.01.2014 12:00:00		20
10	Склад № 2	Вентилятор настольный	Оприходование товаров 000000002 от 15.01.2014 12:00:00		30
11	Склад № 2	Вентилятор оконный	Оприходование товаров 000000002 от 15.01.2014 12:00:00		10
12	Склад № 2	Вентилятор ОРБИТА, STERLING, ЯП	Оприходование товаров 000000002 от 15.01.2014 12:00:00		150
13	Склад № 2	Кондиционер ELEKTA	Оприходование товаров 000000002 от 15.01.2014 12:00:00		44
14					
15					

Рисунок 76

Итак, мы получили остатки на 16.01.2014 по этому регистру. Обратите внимание, как в тексте запроса описывается наложенное условие на виртуальную таблицу (Рисунок 77).

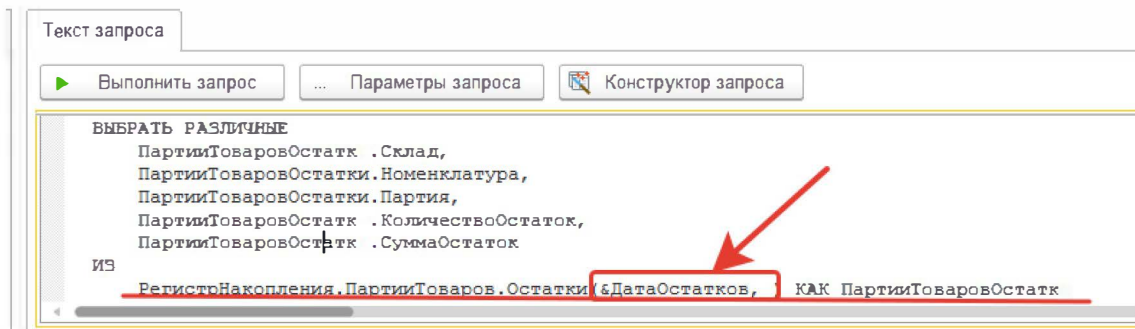


Рисунок 77

То есть период отражается первым параметром виртуальной таблицы, второй не заполнен, так как мы не налагали никаких условий.

1.14 Параметры виртуальной таблицы.

Сейчас потренируемся наложить условие по складу на виртуальную таблицу. Открываем конструктор запроса, переходим в параметры виртуальной таблицы, нажимаем на кнопку выбора условия и перетащим измерение «Склад». В нижнее поле допишем «=&Склад» (Рисунок 78).

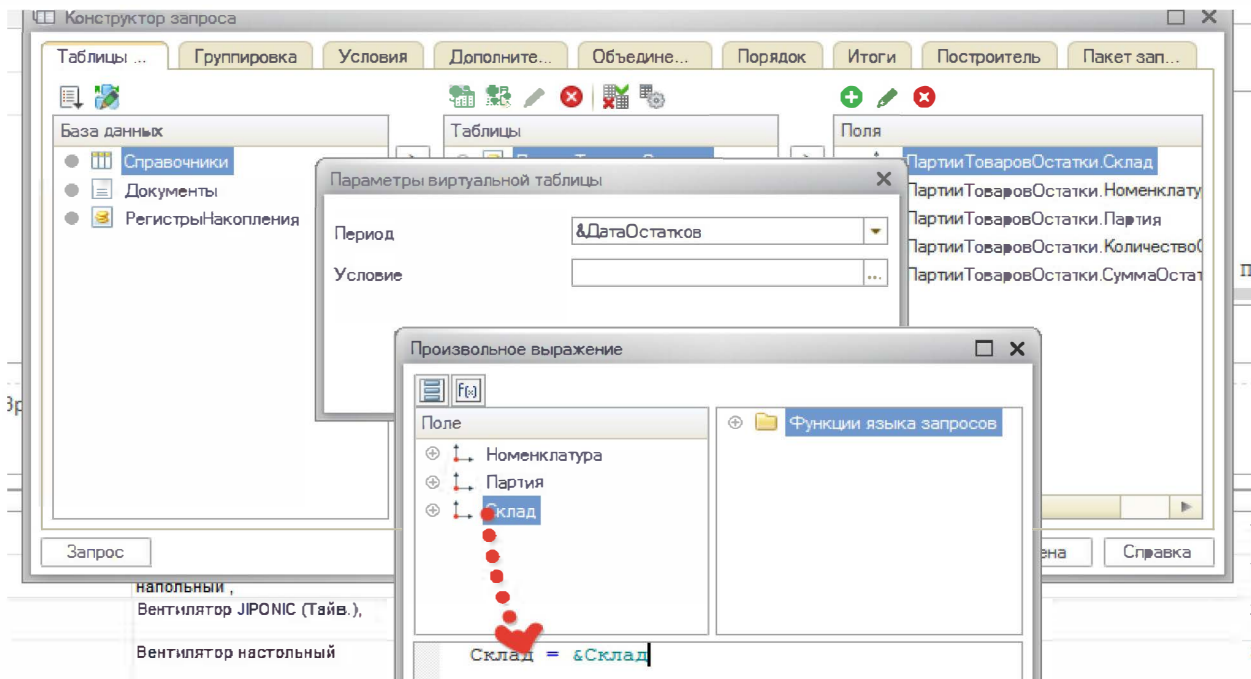


Рисунок 78

Установим параметр склад для запроса (Рисунок 79).

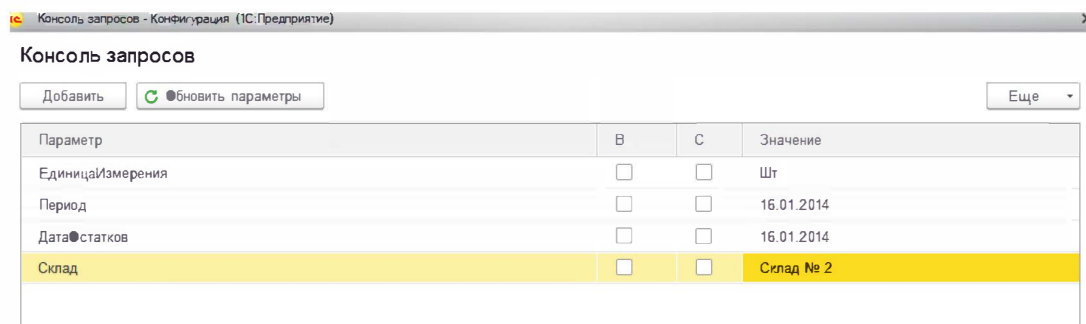


Рисунок 79

Выполним запрос (Рисунок 80).

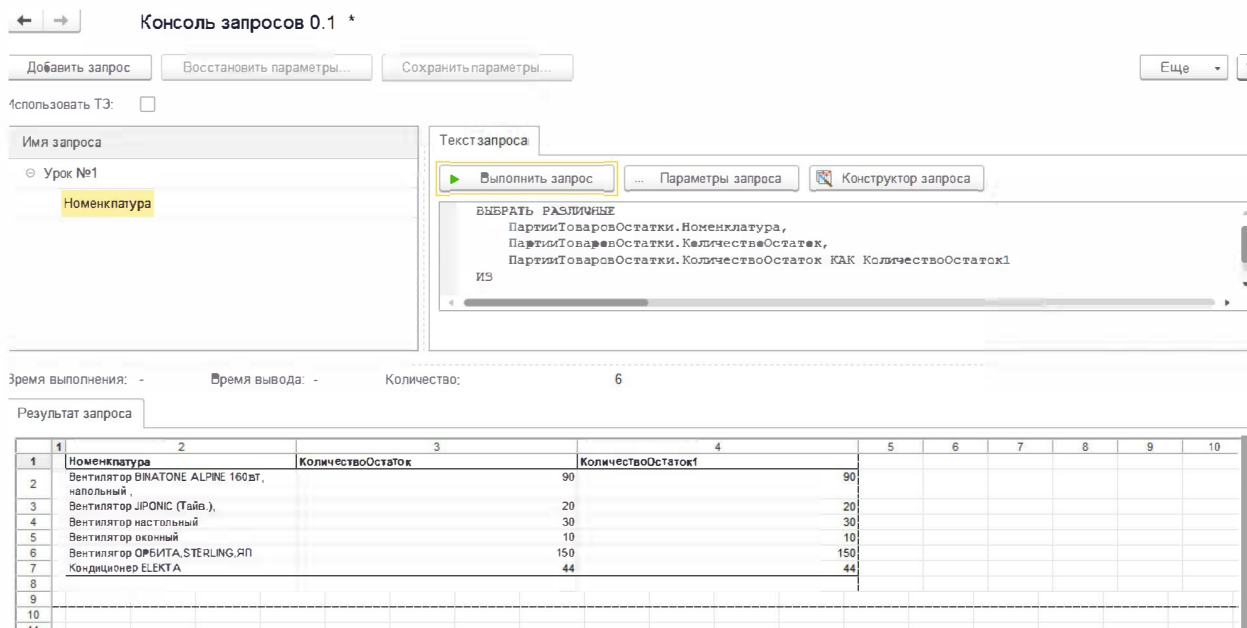


Рисунок 80

Итак, мы получили остатки по складу №2 на 16.01.2014. Еще раз хочу обратить внимание на то, что условия для виртуальных таблиц должны накладываться в параметрах виртуальной таблицы, а не на закладке условия.

Условия могут быть и более сложными. К примеру, нам нужны остатки не только по определенному складу, но и по определенной номенклатуре. Сделаем это (Рисунок 81).

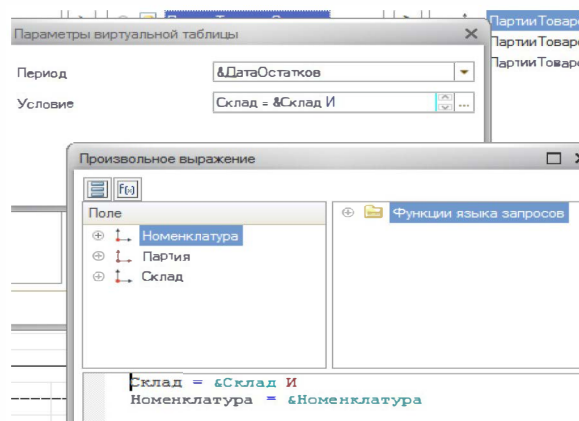


Рисунок 81

То есть мы прямо в окне редактирования условия дописываем оператор «И» и следующее условие по номенклатуре.

Теперь установим параметр и выполним запрос (Рисунок 82-83).

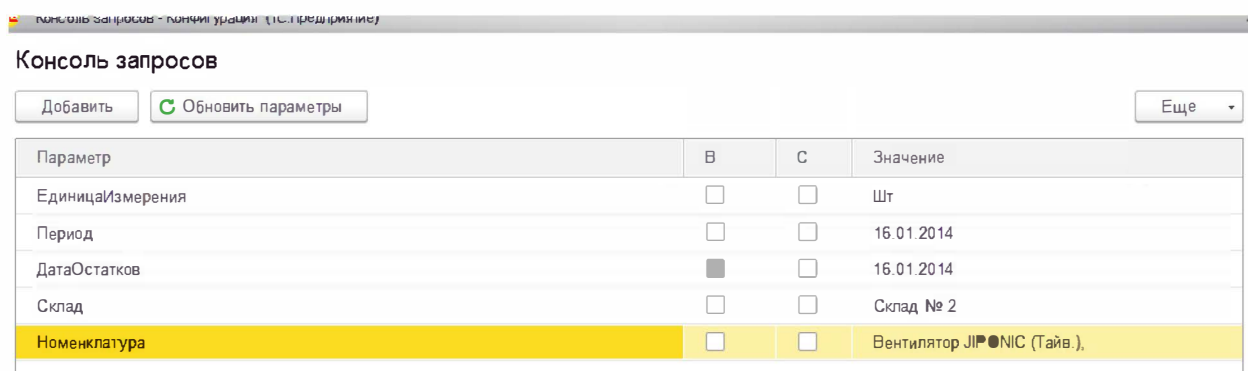


Рисунок 82

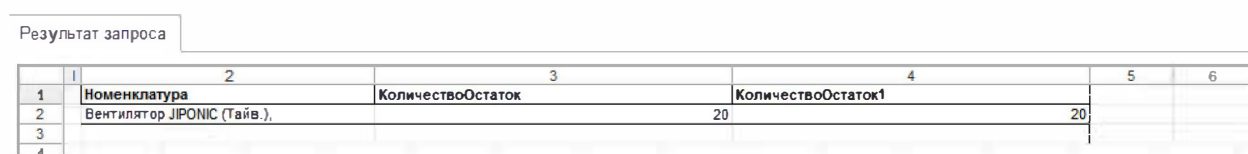


Рисунок 83

Таким образом, мы получили остатки по выбранному складу и номенклатуре.

1.15 Условия. В иерархии.

Познакомимся с логическим оператором языка запросов «В Иерархии», что означает – в подчинении. В случае с номенклатурой, это значит, что будет выбираться номенклатура, находящаяся в определенной группе, а также в группах подчиненных ей.

Для нашего примера с виртуальной таблицей остатков выполним следующие действия: зайдём в конструктор запроса, перейдём в параметры виртуальной таблицы, откроем условие, изменим условие для номенклатуры: вместо оператора «=» перетащим из правого окна логический оператор «В иерархии» и в скобки поместим наш параметр «&Номенклатура» (Рисунок 84-85).

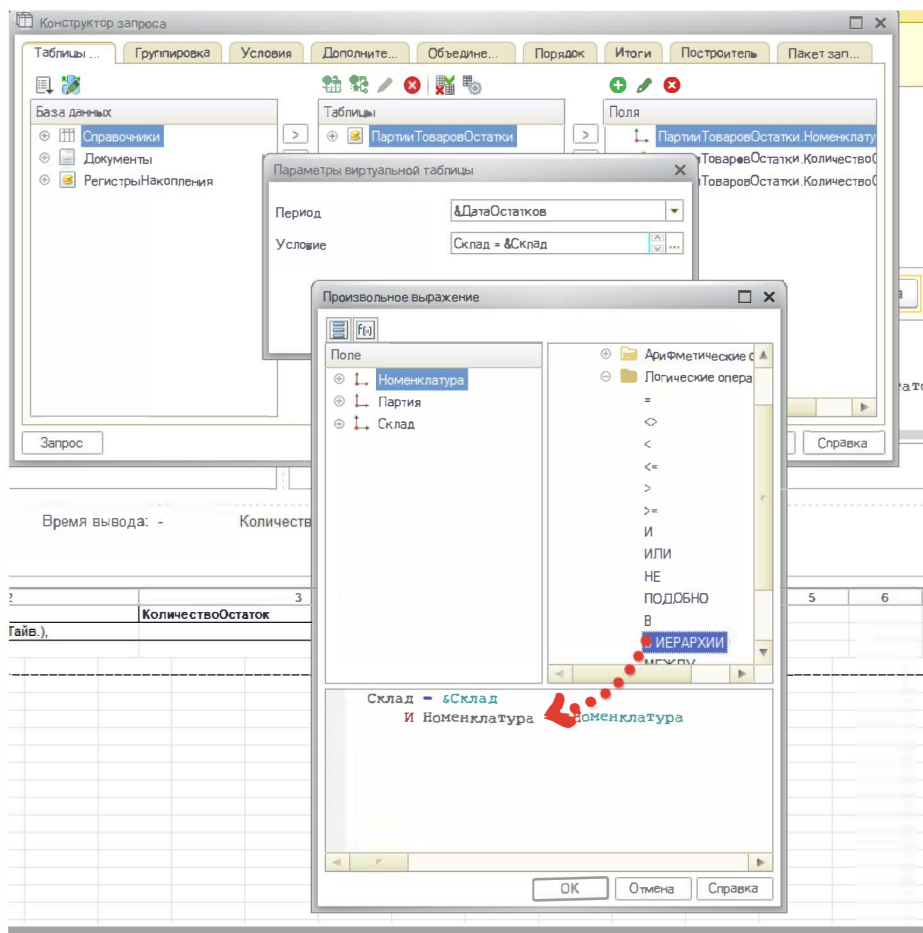


Рисунок 84

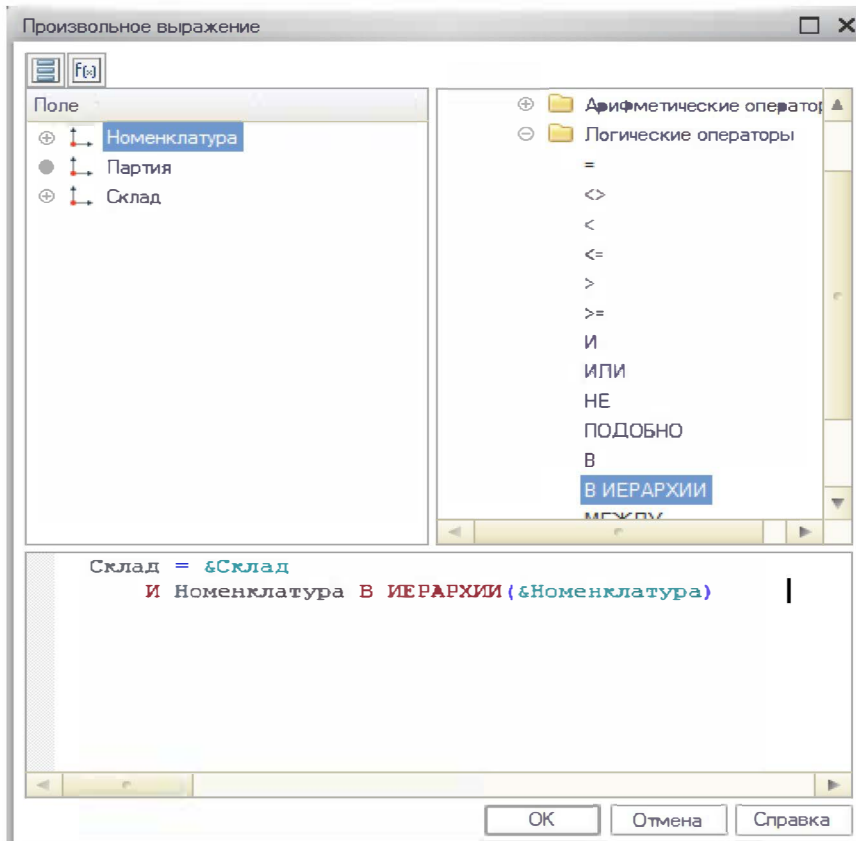


Рисунок 85

Далее установим в качестве параметра группу номенклатуры (Рисунок 86).

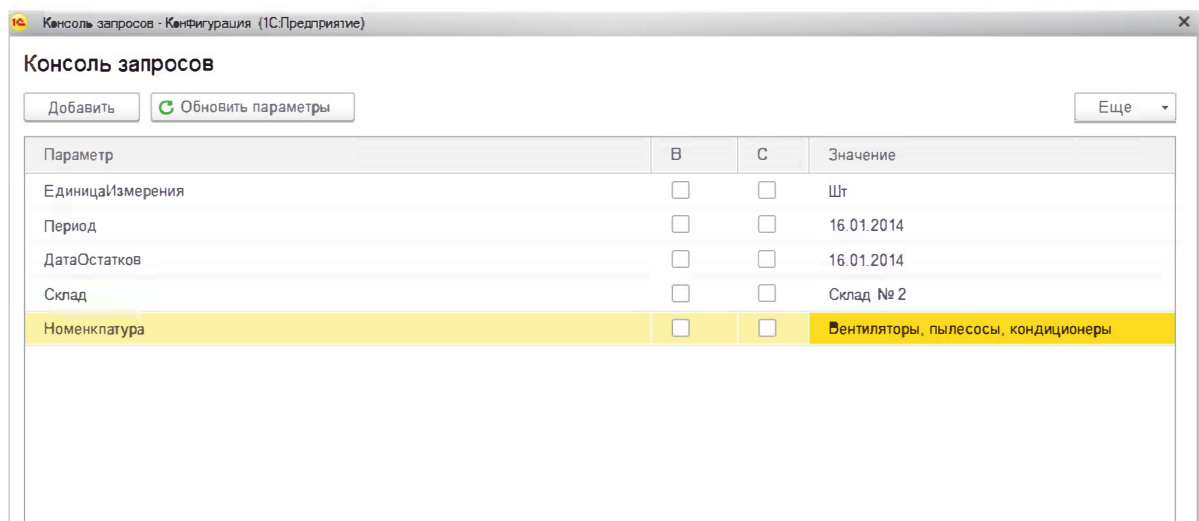


Рисунок 86

Выполним запрос (Рисунок 87).

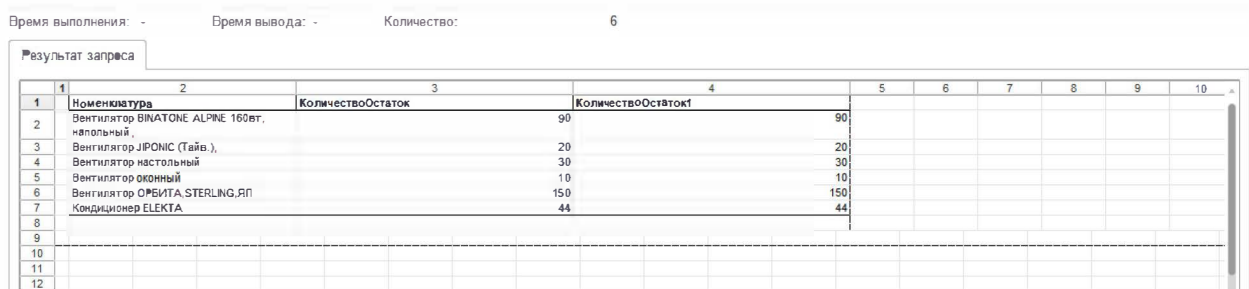


Рисунок 87

В результате выполнения запроса, отобразились остатки на выбранную дату по выбранному складу и по номенклатуре, которая входит в выбранную группу.

Глава 2.

2.1 Обзор конфигурации.

Как и в предыдущем уроке, мы начинаем с обзора нашей конфигурации, а точнее, с подсистем. Для дальнейшей работы необходимо скачать информационную базу, сделать это можно по ссылке https://yadi.sk/d/hpz_R84W8U50nA.

Итак, в нашей конфигурации присутствуют следующие подсистемы (Рисунок 88):

- справочники;
- склад;
- продажи;
- закупки.

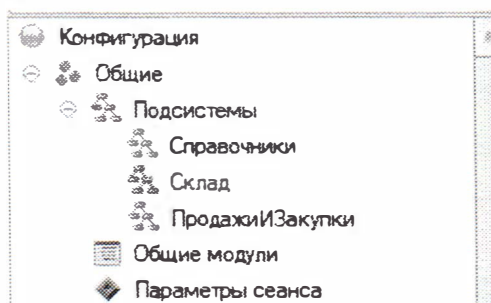


Рисунок 88

Имеется следующий перечень справочников (Рисунок 89):

- номенклатура;
- единицы измерения;
- склады;
- организации;
- контрагенты.

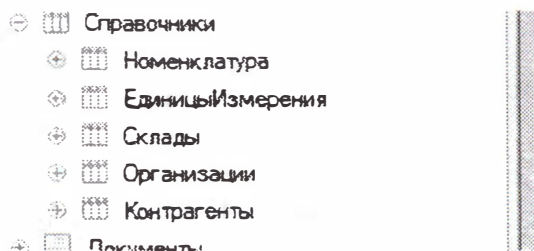


Рисунок 89

Справочник номенклатура состоит из таких реквизитов, как (Рисунок 90):

- единица измерения;
- цена продажная;

- цена закупочная.

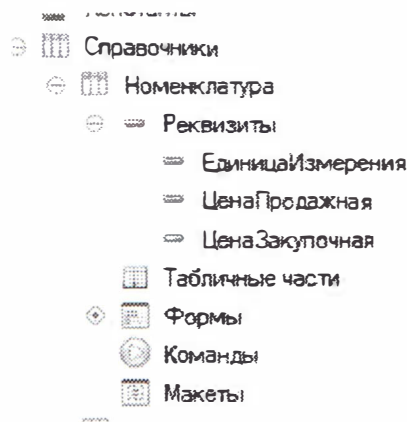


Рисунок 90

Механизм ценообразования намеренно упрощен, дабы не перегружать лишними деталями наш курс. Поэтому цены номенклатуры устанавливаются в каждом элементе справочника.

Документы, в свою очередь, состоят из (Рисунок 91):

- оприходование товаров;
- перемещение товаров;
- списание товаров;
- заказ покупателя;
- реализация товаров и услуг;
- поступление товаров и услуг.

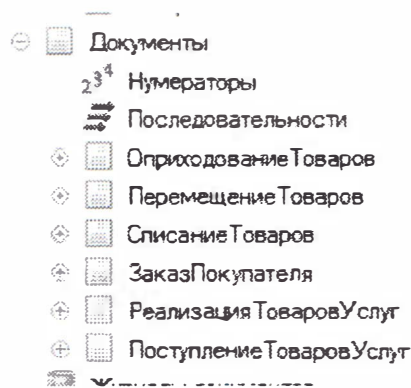


Рисунок 91

Первые три документа являются документами складского контура и делают движения по двум регистрам:

- товары на складах;
- партии товаров.

То есть к регистру «Партии товаров» добавился регистр «Товары на складах». Он имеет измерения: организация, склад, номенклатура и единственный ресурс – количество. У регистра «Партии товаров» учет в разрезе организаций не ведется.

Документ «Заказ покупателя» предназначен для регистрации заявок покупателя и служит основанием для документа «Реализация товаров и услуг». Это, своего рода, документ планирования, делает движения по регистру накопления «Заказы покупателей».

Регистр накопления «Заказы покупателей» состоит из измерений (Рисунок 92):

- организация;
- склад;
- номенклатура;
- заказ.

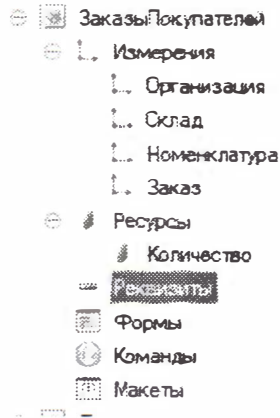


Рисунок 92

Напомню, в регистрах накопления измерения являются разрезами учета.

Документ «Поступление товаров» очень похож на документ «Оприходование товаров», он делает движения по тем же регистрам, а также по регистру «Взаиморасчеты». В свою очередь, регистр накопления «Взаиморасчёты» состоит из измерения «Контрагент» и ресурса «Сумма» (Рисунок 93).

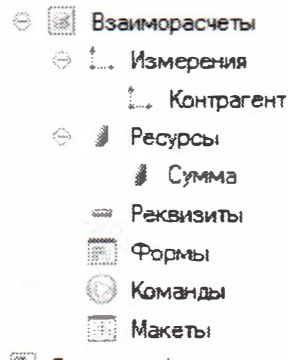


Рисунок 93

Документ «Реализация товаров и услуг» делает движения по всем регистрам. Расходные движения по регистрам: «Партии товаров», «Товары на складах», «Заказы покупателей», «Взаиморасчеты», а также движения по обратному регистру «Продажи».

Регистр накопления «Продажи» состоит из измерений (Рисунок 94):

- организация;
- склад;

– номенклатура.

Ресурсов:

– количество;

– сумма.

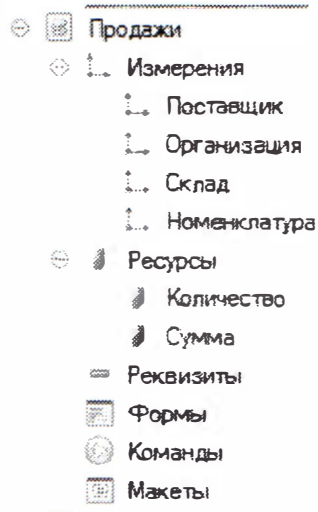


Рисунок 94

Рассмотрим конфигурацию в пользовательском режиме.

Сверху мы видим панель разделов, которая состоит из нескольких разделов.

В том числе, справочники (Рисунок 95):

– организации;

– склады;

– контрагенты;

– номенклатура;

– единицы измерения.

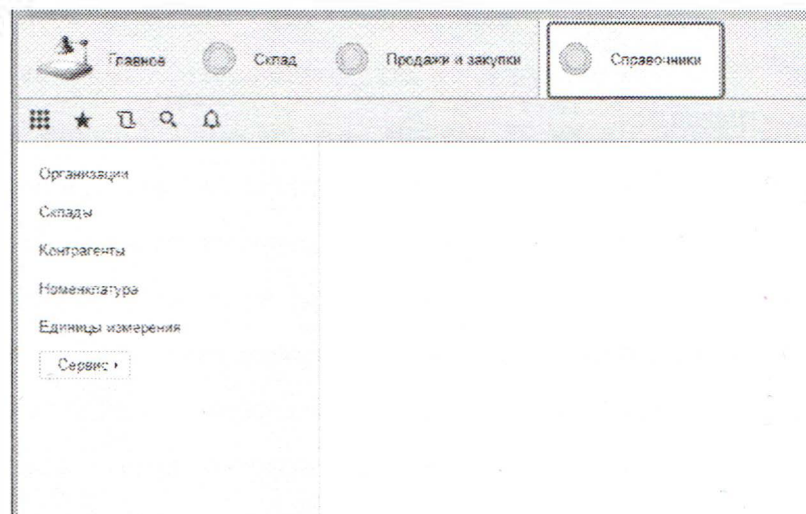


Рисунок 95

Склад (Рисунок 96):

– оприходование товаров;

- перемещение товаров;
- списание товаров;
- склады;
- партии товаров;
- товары на складах.

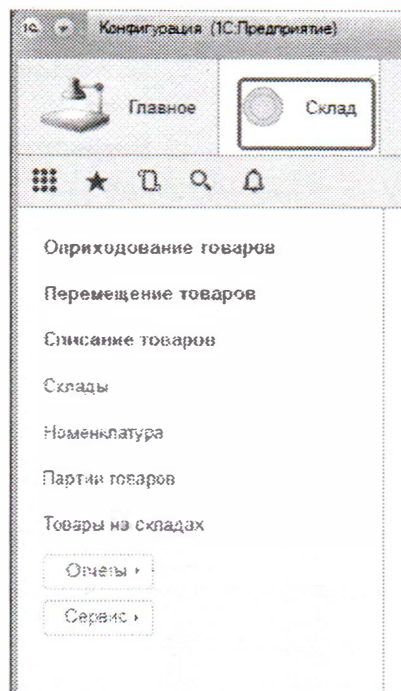


Рисунок 96

Продажи и закупки (Рисунок 97):

- заказ покупателя;
- поступление товаров и услуг;
- реализация товаров и услуг;
- склады;
- контрагенты;
- номенклатура;
- заказы покупателей;
- продажи;
- взаиморасчеты.

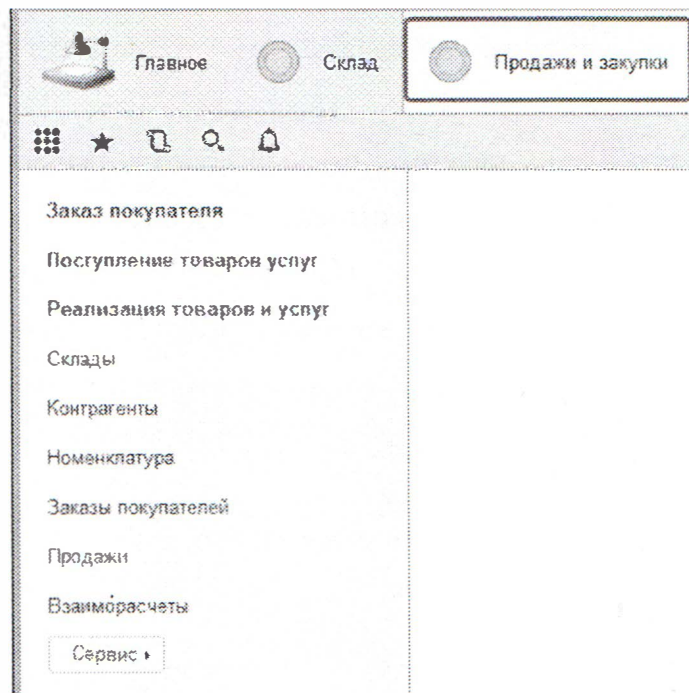


Рисунок 97

Кроме того, в системе введены некоторые документы.

2.2 Таблицы Справочники и Документы.

В этой части мы рассмотрим таблицы справочников и документов. Сделаем мы это при помощи консоли запросов.

Напомню, что справочники и документы, с точки зрения базы данных, являются некими таблицами, хранящими в себе данные. Их реквизиты являются колонками этих таблиц, или в терминах базы данных – полями. Сами элементы справочников и отдельные документы являются строками этих таблиц или, в терминах баз данных, записями. Табличные части есть как у справочников, так и у документов. Что это такое? С точки зрения базы данных, это тоже таблицы, но эти таблицы не имеют объектной сущности. Что это значит? Это значит, что они не существуют сами по себе, то есть все записи в них связаны с каким либо «владельцем»: справочником или документом.

Вообще, говоря, о таблицах баз данных, мы обязаны упомянуть такое понятие как ключевое поле. Ключевое поле определяет уникальность записи, так как каждая запись в базе данных должна быть уникальной. Так вот ключевым полем в таблицах 1С для документов и справочников является ссылка. Ссылка – это некое уникальное значение, которое содержит уникальный идентификатор и тип значения, то есть информацию о том, какому типу принадлежит ссылка, например, к справочнику номенклатура. Уникальный идентификатор генерируется системой при создании объекта и затем не изменяется. Мы

немного углубились в теорию, в надежде, что это поможет в дальнейшем понимании происходящего.

Теперь же перейдем к практике: откроем консоль запросов и запустим конструктор. Перед нами предстают все наши таблицы как объекты конфигурации, в том числе: справочники, документы, регистры накопления. У справочников есть стандартные (обязательные) реквизиты, то есть те, которые создаются независимо от нас, это, например, код и наименование. Есть и реквизиты, которые добавлены разработчиком (Рисунок 98).

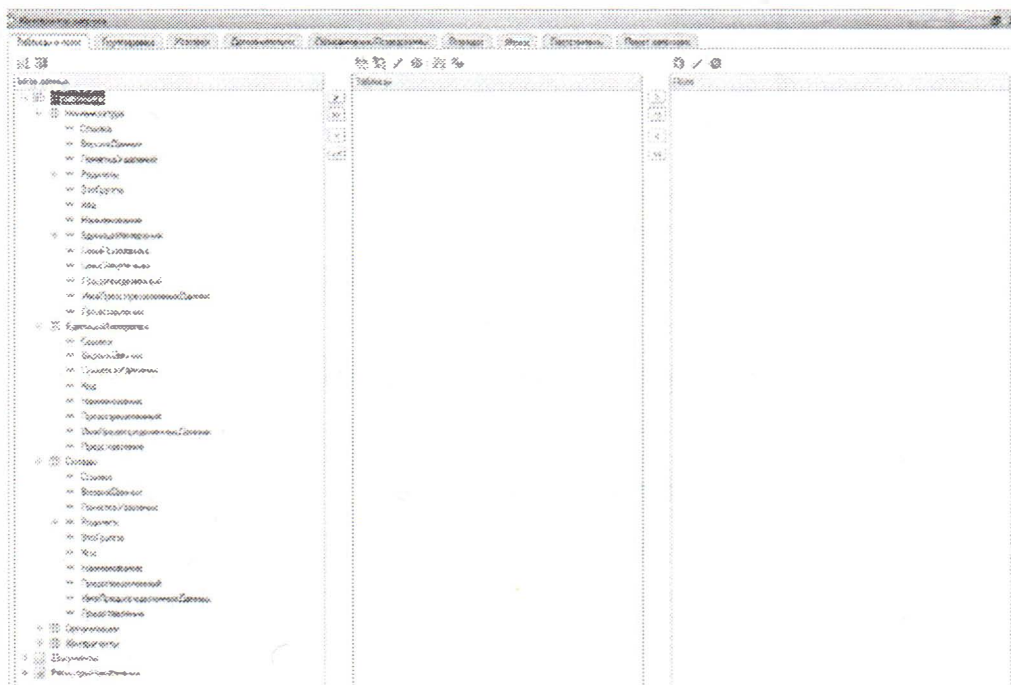


Рисунок 98

Теперь рассмотрим документы.

Документы также имеют некоторые обязательные реквизиты, например, номер и дата. Кроме того, в документах мы видим табличные части, которые упоминали ранее. Документ или справочник может иметь несколько табличных частей, которые также имеют свои реквизиты. В табличной части обязательно есть поле «Ссылка», которое является связующим звеном между документом или справочником и табличной частью. При помощи ссылки мы понимаем, к какому документу или справочнику относится та или иная запись. С табличной частью мы можем работать в качестве отдельной таблицы данных, выбрав ее в качестве источника (Рисунок 99).

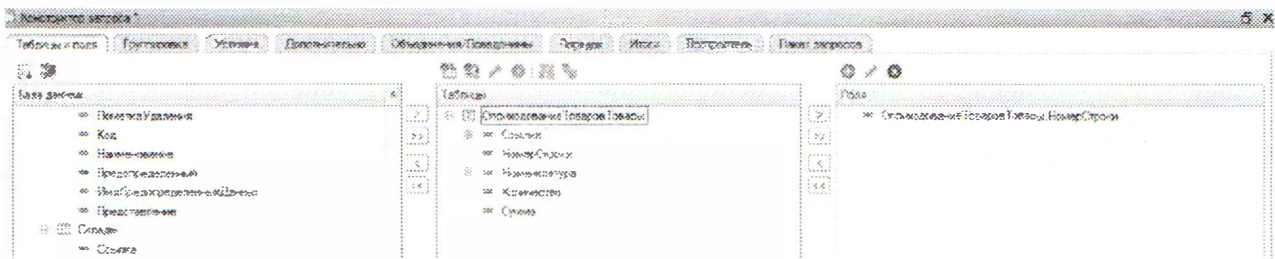


Рисунок 99

Для примера выберем все реквизиты из документа «Оприходование товаров». Табличную часть «Товары» выбирать не будем. Посмотрим на результат запроса (Рисунок 100).

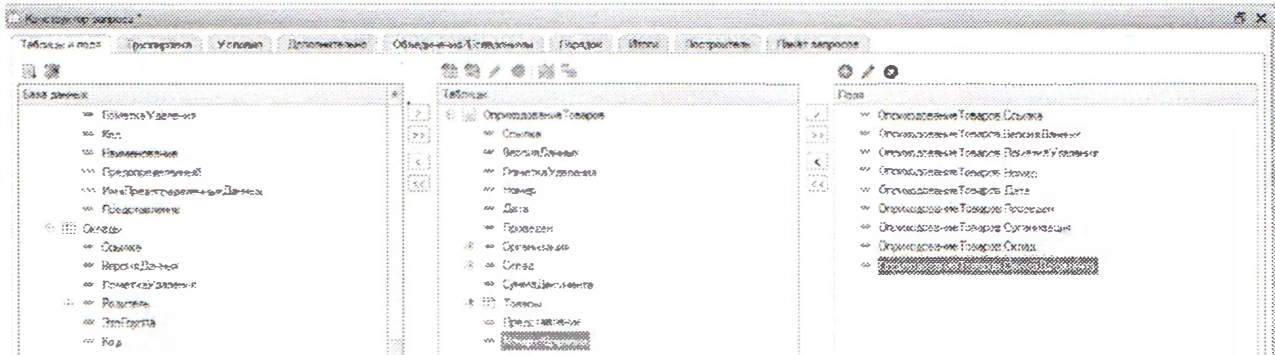


Рисунок 100

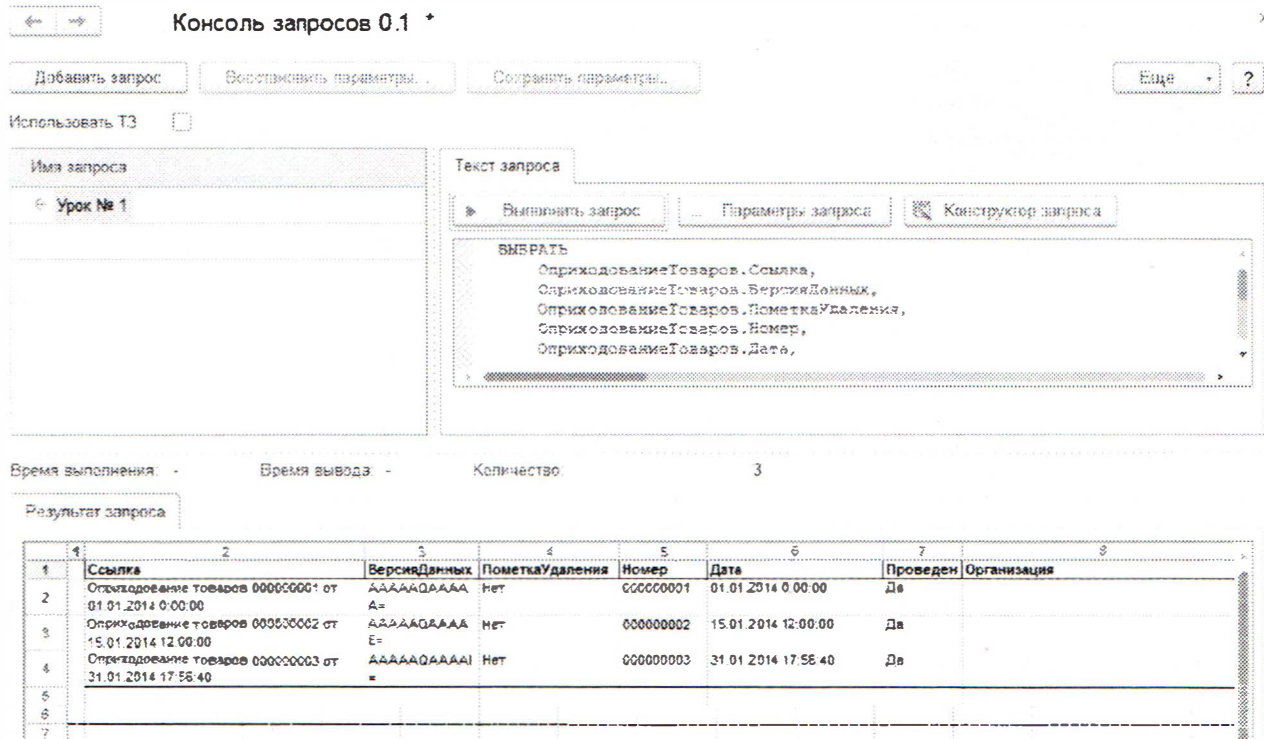


Рисунок 101

Теперь вместо документа «Оприходование товаров» выберем табличную часть «Товары».

Посмотрим на содержимое табличных частей всех документов (Рисунок 102-103).

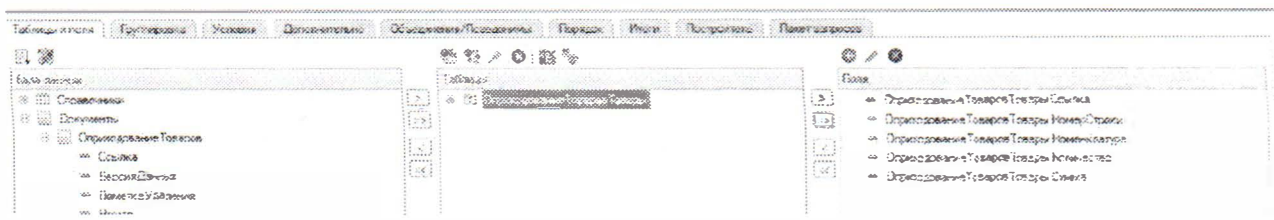


Рисунок 102

ссылка: Т3

Имя запроса: Урок № 1

Текст запроса:

▶ Выполнить запрос

Параметры запроса

Конструктор запроса

ГЕНЕРАТЬ

ОприходованиеТоваровТовары.Ссылка,

ОприходованиеТоваровТовары.НомерСтроки,

ОприходованиеТоваровТовары.Номенклатура,

ОприходованиеТоваровТовары.Количество,

ОприходованиеТоваровТовары.Сумма

время выполнения: - время вывода: - количество: 18

Результат запроса

1	2	3	4	5	6	7	8	9
№	Ссылка	НомерСтроки	Номенклатура	Количество	Сумма			
2	Оприходование товаров 000000001 от 01.01.2014 0:00:00	6	Кондиционер ELEKTA	44	198 000,00			
3	Оприходование товаров 000000001 от 01.01.2014 0:00:00	3	Вентилятор настольный	30	9 000,00			
4	Оприходование товаров 000000001 от 01.01.2014 0:00:00	1	Вентилятор BRMATONE ALPINE 160вт. настольный.	90	117 000,00			
5	Оприходование товаров 000000001 от 01.01.2014 0:00:00	5	Вентилятор ОРБИТА STERLING,ЯП.	150	60 000,00			
6	Оприходование товаров 000000001 от 01.01.2014 0:00:00	4	Вентилятор оконный	10	4 000,00			
7	Оприходование товаров 000000001 от 01.01.2014 0:00:00	2	Вентилятор JIPONIC (Таив.)	30	171 000,00			
8	Оприходование товаров 000000002 от 15.01.2014 12:00:00	5	Вентилятор ОРБИТА STERLING,ЯП.	100	40 000,00			
9	Оприходование товаров 000000002 от 15.01.2014 12:00:00	2	Вентилятор JIPONIC (Таив.)	32	60 000,00			

Рисунок 103

2.3 Группировки.

В этой части поговорим о группировке в запросах. Для этого немного изменим текст запроса. Уберем поля «ссылка» и «номер строки», выполним запрос. Видим информацию обо всей когда-либо оприходованной номенклатуре. При этом информация об одной и той же номенклатуре может встречаться не один раз, что вполне естественно, так как одна и та же номенклатура могла быть оприходована в разных документах в разное время.

Номенклатура	Количество	Сумма
Кондиционер ELEKTA	44	136 000,00
Вентилятор настольный	30	9 000,00
Вентилятор BINATONE ALPINE 160вт. напо .	90	117 000,00
Вентилятор ОРБИТА.STERLING.ЯП	150	50 000,00
Вентилятор оконный	10	4 000,00
Вентилятор JIPONIC (Тайв .)	90	171 000,00
Вентилятор ОРБИТА.STERLING.ЯП.	100	40 000,00
Вентилятор JIPONIC (Тайв .)	32	50 800,00
Вентилятор настольный	40	12 000,00
Вентилятор BINATONE ALPINE 160вт. напо .	94	122 200,00
Кондиционер ELEKTA	20	90 000,00
Вентилятор оконный	15	6 000,00
Вентилятор JIPONIC (Тайв .)	80	152 000,00

Рисунок 104

А что, если нам необходимо получить итоговую информацию по каждой позиции? Как нам этого добиться? Это и есть группировка в запросе. Для этого в консоли запросов существует специальная закладка, где указываются групповые и суммируемые поля, по которым будет вычислена сумма или другая агрегатная функция, например, максимум. Агрегатные функции – это такие функции, которые принимают на вход множество различных значений и выдают одно результирующее значение. Мы часто пользуемся агрегатными функциями в жизни, не подозревая об этом. В общем, если б Остап узнал, что он играет такие мудреные партии и сталкивается с такой испытанной защитой, он крайне бы удивился.

Примеры агрегатных функций:

- минимум;
- максимум;
- количество (количество записей);
- среднее;
- количество различных (количество различных записей).

В нашем случае необходима итоговая информация по сумме и количеству, поэтому нам требуется вычислить сумму при группировке.

Откроем конструктор запросов, перейдем на вкладку «Группировка», в качестве группового поля укажем номенклатуру, а в качестве суммируемых полей укажем количество и сумму. Для выбора группового поля достаточно дважды щелкнуть по нему левой кнопкой мыши, для выбора суммируемого поля необходимо выделить его и нажать кнопку переноса напротив соответствующего раздела (Рисунок 105-106).

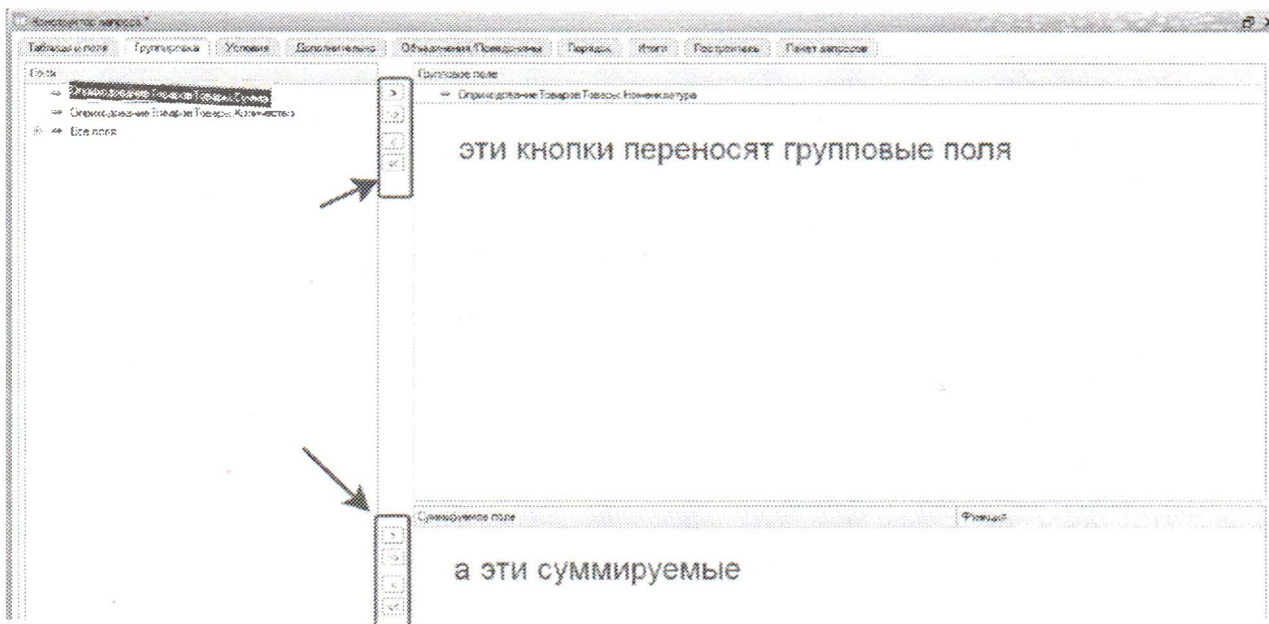


Рисунок 105

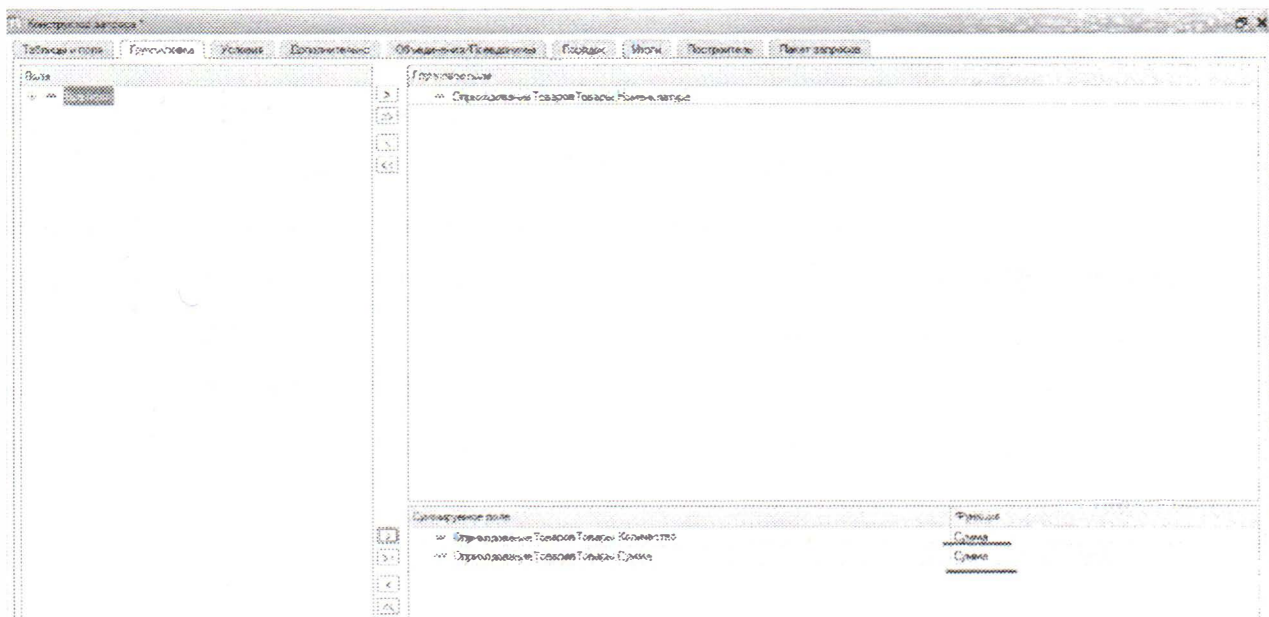


Рисунок 106

Посмотрим, как изменился текст запроса. В нем появились новые ключевые слова «Сумма» и «Сгруппировать по» (Рисунок 107).


```

ВЫБРАТЬ
    ОприходованиеТоваров.Товары.Номенклатура,
    СУММА (ОприходованиеТоваров.Товары.Количество) КАК Количество,
    СУММА (ОприходованиеТоваров.Товары.Сумма) КАК Сумма
ИЗ
    Документ.ОприходованиеТоваров.Товары КАК ОприходованиеТоваров.Товары

СГРУППИРОВАТЬ ПО
    ОприходованиеТоваров.Товары.Номенклатура

```

Рисунок 107

Ключевое слово «Сумма» является агрегатной функцией языка запросов и в качестве параметров принимает поле, которое требуется суммировать. Поэтому после слова «Сумма» поле заключено в скобки, а ключевое слово «Сгруппировать по» предшествует списку групповых полей. В нашем случае групповое поле одно, но их может быть несколько.

Каков же результат выполнения запроса? Мы видим, что повторяющихся строк не стало, а значения суммы и количества для одной и той же номенклатуры суммировались. Общее количество строк уменьшилось (Рисунок 108).

Результат запроса

1	2	3	4	5
1	Номенклатура	Количество	Сумма	
2	Кондиционер ELEKTA	104	468 000,00	
3	Вентилятор настольный	114	34 200,00	
4	Вентилятор BRATONE ALPINE 160вт, напольный	206	267 800,00	
5	Вентилятор ОРБИТА, STERLING, ЯП.	280	112 000,00	
6	Вентилятор сконный	85	34 000,00	
7	Вентилятор JIPONIC (Тайв.)	202	383 800,00	
8				
9				

Рисунок 108

Необходимо отметить, что группировка важна только для реальных таблиц. В виртуальных таблицах все данные уже сгруппированы, поэтому группировать их не имеет особого смысла. Сгруппированы они, конечно же, по измерениям.

Для примера сделаем запрос к реальной таблице регистра накопления «Товары на складах» (Рисунок 109-110).

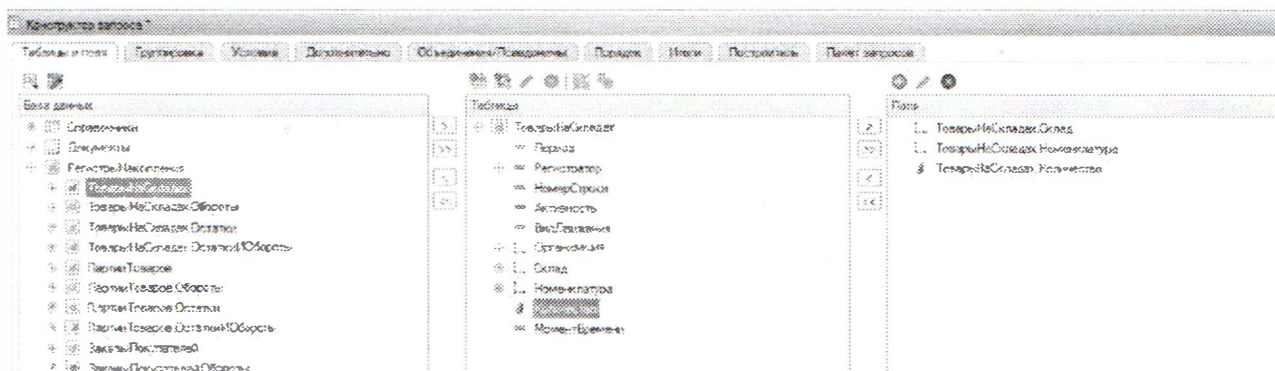


Рисунок 109

Результат запроса

1	2	3	4
1	Склад	Номенклатура	Количество
2	Склад № 1	Вентилятор настольный	150
3	Склад № 1	Кондиционер ELEKTA	27
4	Склад № 1	Вентилятор оконный	100
5	Склад № 1	Вентилятор оконный	10
6	Склад № 1	Вентилятор VENTONE ALPINE 150вт, напольный	90
7	Склад № 1	Вентилятор JIPONIC (Тайв)	90
8	Склад № 1	Вентилятор настольный	30
9	Склад № 1	Вентилятор оконный	10
10	Склад № 1	Вентилятор ОРБИТА, STERLING ЯП	150
11	Склад № 1	Кондиционер ELEKTA	44
12	Склад № 2	Вентилятор VENTONE ALPINE 150вт, напольный	94
13	Склад № 2	Вентилятор JIPONIC (Тайв)	32
14	Склад № 2	Вентилятор настольный	40
15	Склад № 2	Вентилятор оконный	15
16	Склад № 2	Вентилятор ОРБИТА, STERLING ЯП	100
17	Склад № 2	Кондиционер ELEKTA	20
18	Склад № 1	Вентилятор VENTONE ALPINE 150вт, напольный	22
19	Склад № 1	Вентилятор JIPONIC (Тайв)	80
20	Склад № 1	Вентилятор настольный	44
21	Склад № 1	Вентилятор оконный	80
22	Склад № 1	Вентилятор ОРБИТА, STERLING ЯП	30
23	Склад № 1	Кондиционер ELEKTA	40
24	Склад № 1	Вентилятор настольный	5
25	Склад № 3	Вентилятор настольный	5
26	Склад № 1	Телевизор "СМД РР"	15

Рисунок 110

Теперь сгруппируем их по складу и номенклатуре. Это значит, что данные будут сгруппированы по номенклатуре для каждого склада (Рисунок 111-112).

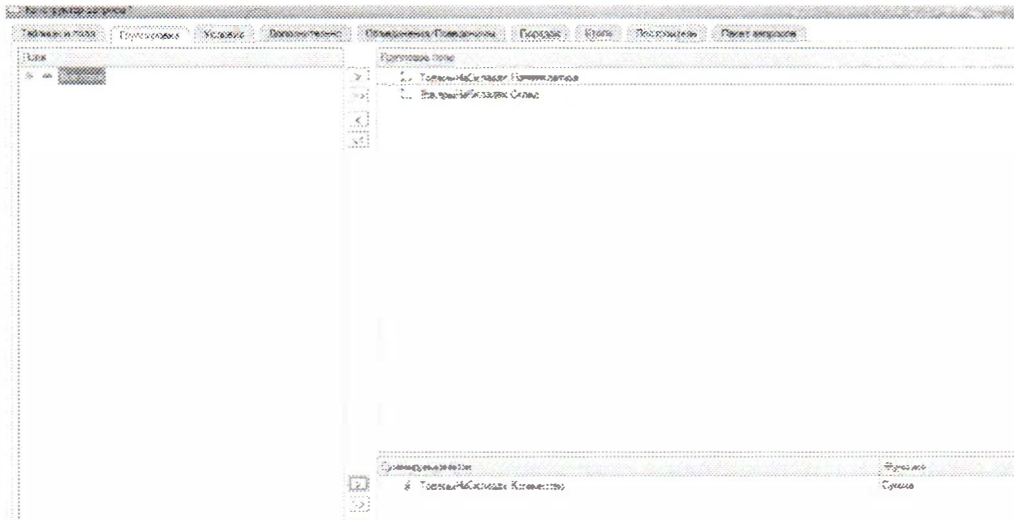


Рисунок 111

Результат запроса				
1	2	3	4	
4	Склад № 1	Вентилятор оконный		180
5	Склад № 1	Вентилятор BINATONE ALPINE 160вт, напольный		112
6	Склад № 1	Вентилятор JIPONIC (Тайв.)		170
7	Склад № 1	Вентилятор ОРБИТА, STERLING, ЯП.		180
8	Склад № 2	Вентилятор BINATONE ALPINE 160вт, напольный		94
9	Склад № 2	Вентилятор JIPONIC (Тайв.)		32
10	Склад № 2	Вентилятор настольный		40
11	Склад № 2	Вентилятор оконный		15
12	Склад № 2	Вентилятор ОРБИТА, STERLING, ЯП.		100
13	Склад № 2	Кондиционер ELEKTA		20
14	Склад № 3	Вентилятор настольный		5
15	Склад № 1	Телевизор "SHARP"		25

Рисунок 112

Мы видим, что одна и та же позиция сгруппирована и для склада №2, и для склада №3, так как группировка задана по складу и номенклатуре.

Теперь повторим то же самое с виртуальной таблицей остатков. Группировку, при этом, выполнять не станем (Рисунок 113-114).

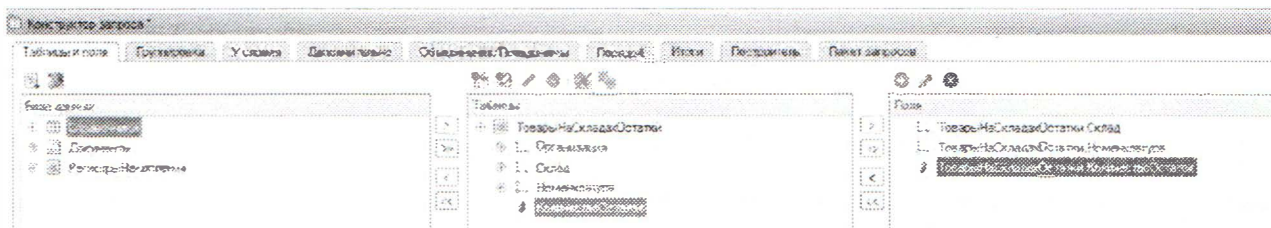


Рисунок 113

Время выполнения: - Время вывода: - Количество: 16

Результат запроса				
1	2	3	4	5
3	Склад № 1	напольный,		
4	Склад № 1	Вентилятор JIPONIC (Тайв.)		170
5	Склад № 1	Вентилятор настольный		219
6	Склад № 1	Вентилятор оконный		180
7	Склад № 1	Вентилятор ОРБИТА, STERLING, ЯП.		180
8	Склад № 1	Кондиционер ELEKTA		111
9	Склад № 2	Вентилятор BINATONE ALPINE 160вт, напольный		94
10	Склад № 2	Вентилятор JIPONIC (Тайв.)		32
11	Склад № 2	Вентилятор настольный		40
12	Склад № 2	Вентилятор оконный		15
13	Склад № 2	Вентилятор ОРБИТА, STERLING, ЯП.		100
14	Склад № 2	Кондиционер ELEKTA		20
15	Склад № 1	СТИНОЛ 101		22
16	Склад № 1	Телевизор "JVC"		15
17	Склад № 1	Телевизор "SHARP"		25
18	Склад № 3	Вентилятор настольный		5

Рисунок 114

Как видим, результат такой же.

2.4 Итоги в запросе.

В данной части нам предстоит узнать об итогах в запросе.

Вот ведь, скажете вы, группировки то мы только что рассмотрели, и там были итоговые записи, а здесь что? Ну, скажем так, вы почти правы. Итоги отличаются от 59 Верный старт в 1С www.work-1c.ru 88002344322 Бесплатный звонок

группировки только тем, что появляется дополнительная строка с итогами по выбранным группировочным полям, то есть разница с группировками состоит только в том, что группировки «сворачивают» строки (уменьшают их количество) с одинаковыми групповыми полями, а итоги добавляют итоговую строку по группировочным полям, при этом, если группировку не применять, то строки могут повторяться, но появятся дополнительные строки с итогами. Звучит, может и не очень понятно, поэтому, к делу – рассмотрим все на картинках.

Предлагаю ничего не менять в нашем запросе, а лишь добавить итоги по складу, то есть «склад» будет группировочным полем, а итоговым полем будет «количество» (Рисунок 115).

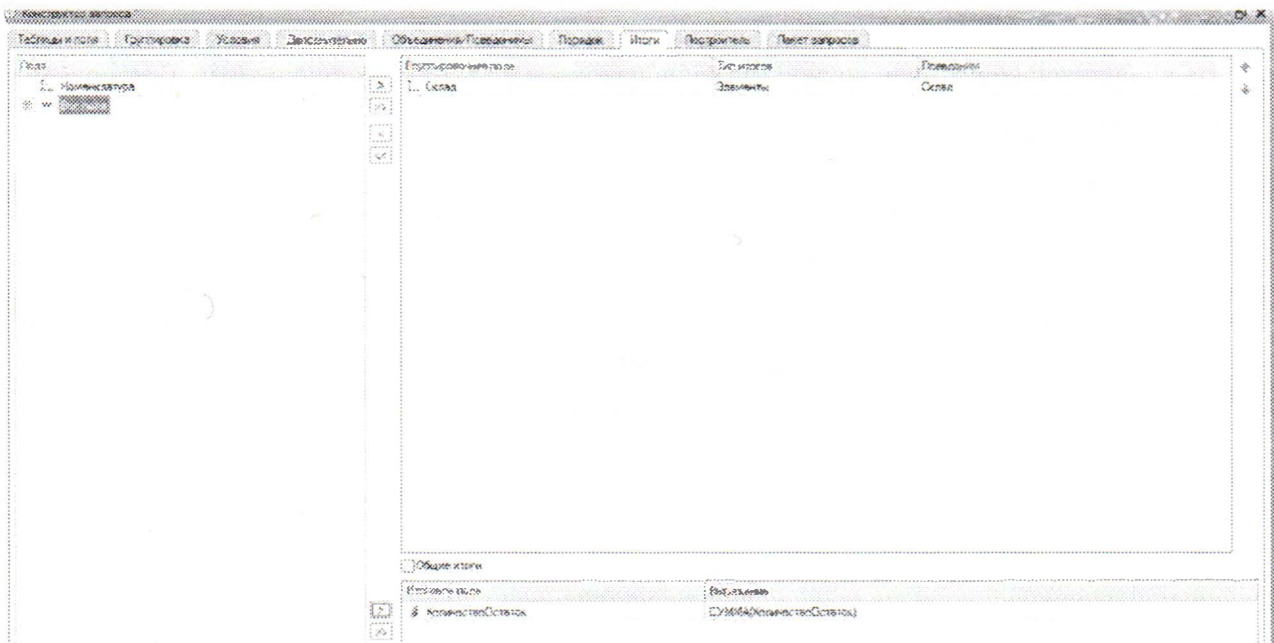


Рисунок 115

Посмотрим, как изменился запрос (Рисунок 116).

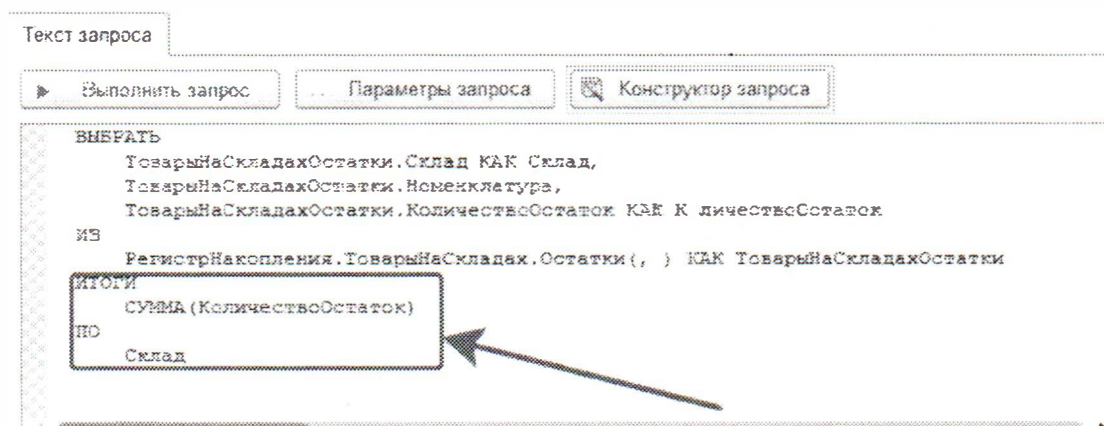


Рисунок 116

Можем увидеть, что появилась новая секция – «Итоги». Обратите внимание, что после слова «Итоги» сначала перечисляются агрегатные функции с полями, в качестве

параметров, а затем ключевое слово «По», после которого через запятую перечисляются группировочные поля. Отметим, что после последнего поля запятая не ставится.

Теперь посмотрим, чем отличается результата выполнения запроса от прежнего (Рисунок 117).

Результат запроса

1	2	3	4	5
1	Склад	Номенклатура	КоличествоОстаток	
2	Склад № 1			1 634
3	Склад № 1	Вентилятор BINATONE ALPINE 160вт, напольный		112
4	Склад № 1	Вентилятор JIPONIC (Тайе.)		170
5	Склад № 1	Вентилятор настольный		219
6	Склад № 1	Вентилятор оконный		180
7	Склад № 1	Вентилятор ОРБИТА, STERLING, ЯП.		183
8	Склад № 1	Кондиционер ELEKTA		111
9	Склад № 1	СТИНОЛ 121		22
10	Склад № 1	Телевизор "JVC"		15
11	Склад № 1	Телевизор "SHARP"		25
12	Склад № 2			304
13	Склад № 2	Вентилятор BINATONE ALPINE 160вт, напольный		94
14	Склад № 2	Вентилятор JIPONIC (Тайе.)		32
15	Склад № 2	Вентилятор настольный		40
16	Склад № 2	Вентилятор оконный		15
17	Склад № 2	Вентилятор ОРБИТА, STERLING, ЯП.		100
18	Склад № 2	Кондиционер ELEKTA		20
19	Склад № 3			5
20	Склад № 3	Вентилятор настольный		5
21				

Рисунок 117

Стрелочками и подчеркиванием показаны итоговые строки – те самые дополнительные строки, о которых мы говорили ранее. Итоги выведены по каждому складу. В графе «КоличествоОстаток» отражена сумма всех товаров, которые имеются на складе по количеству. Графа «Номенклатура» не заполнена, так как номенклатура не является ни суммируемым, ни группировочным полем. В этом поле значение равно NULL – это особое значение, которое обозначает ничего, пустоту, то есть NULL не равен ни нулю, ни пустой ссылке, ни пустой строке, ни чему бы то ни было.

Теперь немного изменим наш запрос и выведем итоги по номенклатуре.

Для этого откроем конструктор запроса и перейдем на закладку «Итоги». Далее удалим итоговое поле «Склад», а вместо него перенесем поле «Номенклатура» (Рисунок 118).

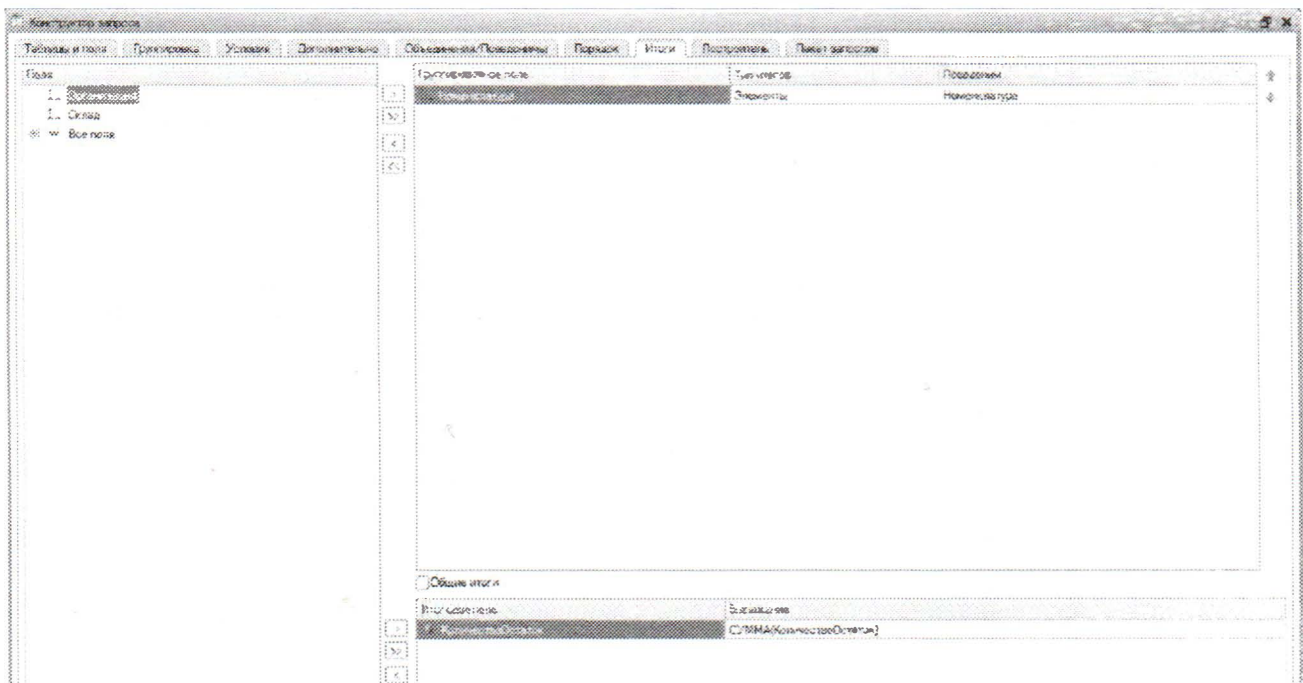


Рисунок 118

Итак, мы получили итоги по номенклатуре на всех складах (Рисунок 119).

Т3Результата

Стрелками указаны итоговые строки

Организация	Склад	Номенклатура	Количество	Итого
		Вентилятор BINATONE ALPIN	164	
	Склад № 1	Вентилятор BINATONE ALPIN	90	
	Склад № 2	Вентилятор BINATONE ALPIN	54	
		Вентилятор JIPONIC (Тайв.)	122	
	Склад № 1	Вентилятор JIPONIC (Тайв.)	90	
	Склад № 2	Вентилятор JIPONIC (Тайв.)	32	
		Вентилятор настольный	220	
	Склад № 1	Вентилятор настольный	36	
	Склад № 2	Вентилятор настольный	40	
ООО "Торговый дом"	Склад № 1	Вентилятор настольный	160	
		Вентилятор оконный	135	

Рисунок 119

Что мы можем сделать с информацией, в которой есть итоги? Дело в том, что при программной работе с запросом мы можем получить таблицу значений. Она является как бы "плоской", одноуровневой, то есть строку с итогами мы не сможем отличить от строки детальной записи.

2.5 Копирование текста запроса.

Эта часть призвана научить вас, дорогие читатели, такому приему разработки, как копирование текста запроса.

Текст запроса можно копировать прямо из конструктора запроса и вставлять в конструктор запроса. Зачем это нужно, спросите вы? Затем, что когда мы ведем разработку,

то зачастую работаем с запросом во встроенном языке, а отладку запроса производим в консоли запросов в режиме «1С:Предприятие».

Вы уже видели ранее, что в коде текст запроса оформлен как строка и снабжен символами переноса строки (Рисунок 120).

```
Запрос = Новый Запрос;  
Запрос.Текст = "ВЫБРАТЬ  
    | Номенклатура.Наименование,  
    | Номенклатура.Код,  
    | Номенклатура.ЕдиницаИзмерения  
    | ИЗ  
    | Справочник.Номенклатура КАК Номенклатура  
    | ГДЕ  
    | Номенклатура.ЕдиницаИзмерения = &ЕдиницаИзмерения" ;
```

Рисунок 120

Благодаря этим символам, взять текст запроса из консоли и напрямую вставить его в программный код, не получится. Его придется дополнительно редактировать, что может сказаться на удобстве и производительности.

Откройте конфигуратор и посмотрите на отчет «ОстаткиТоваровНаСкладах». Это своего рода шаблон, в котором есть макет табличного документа для вывода отчета и форма самого отчета.

Весь код отчета располагается в модуле формы. Обратите внимание, что в программном коде нет текста запроса (Рисунок 121).

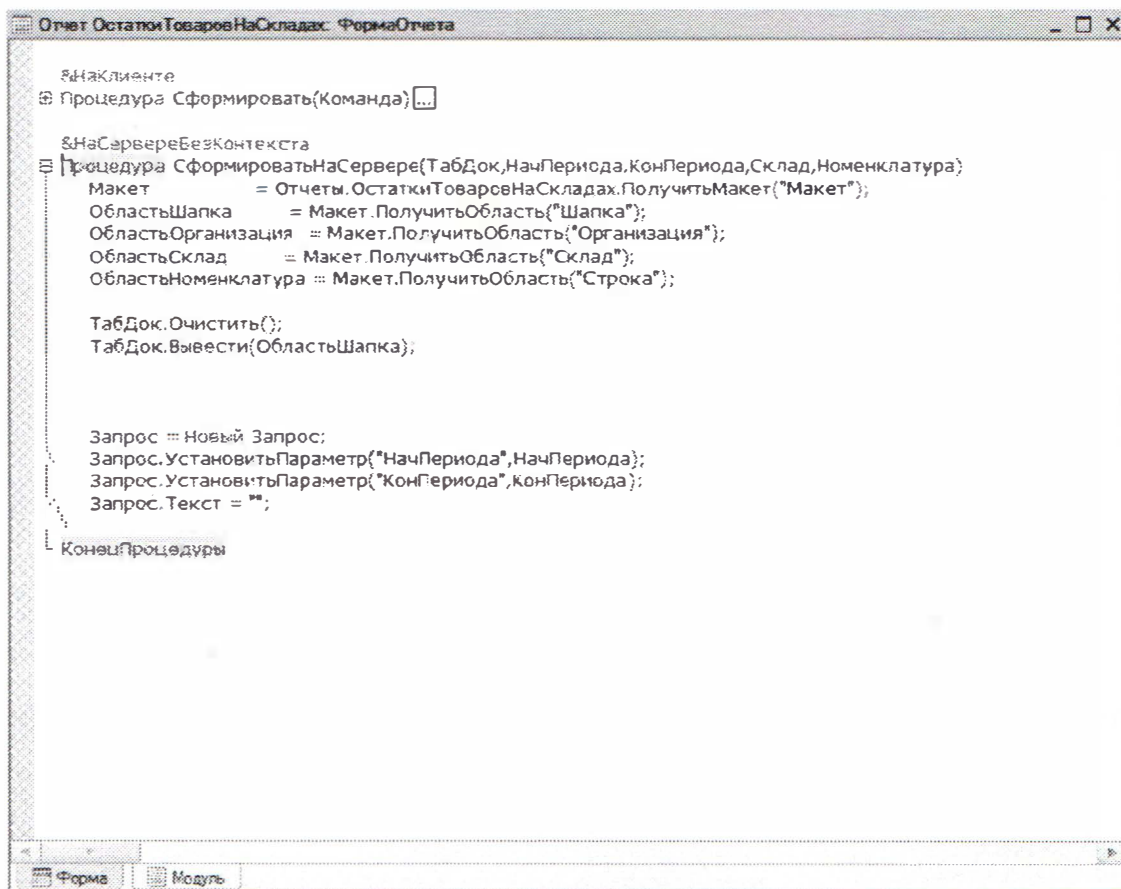


Рисунок 121

Мы планируем скопировать его туда из консоли запросов.

Сначала в консоли запросов немного изменим запрос, с которым мы работали в предыдущей части: добавим поле «Организация» в выходные поля нашего запроса, установим параметр периода получения остатков (&ДатаОстатка) для виртуальной таблицы, для поля «КоличествоОстаток» установим псевдоним «Количество» (Рисунок 122).

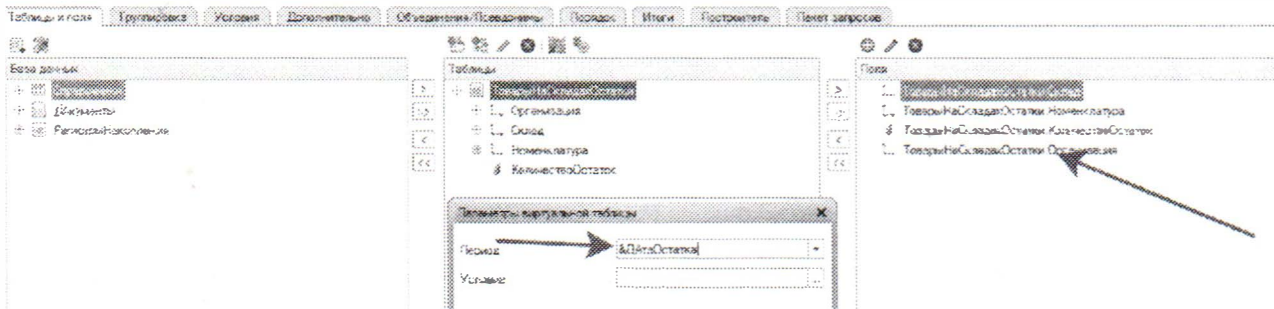


Рисунок 122

Теперь попробуем скопировать запрос из консоли.

Для этого в конструкторе запроса нажмем на кнопку «Запрос». Перед нами откроется окно с текстом запроса, который мы можем скопировать, а если нажмем кнопку редактирования, то и редактировать (Рисунок 123).

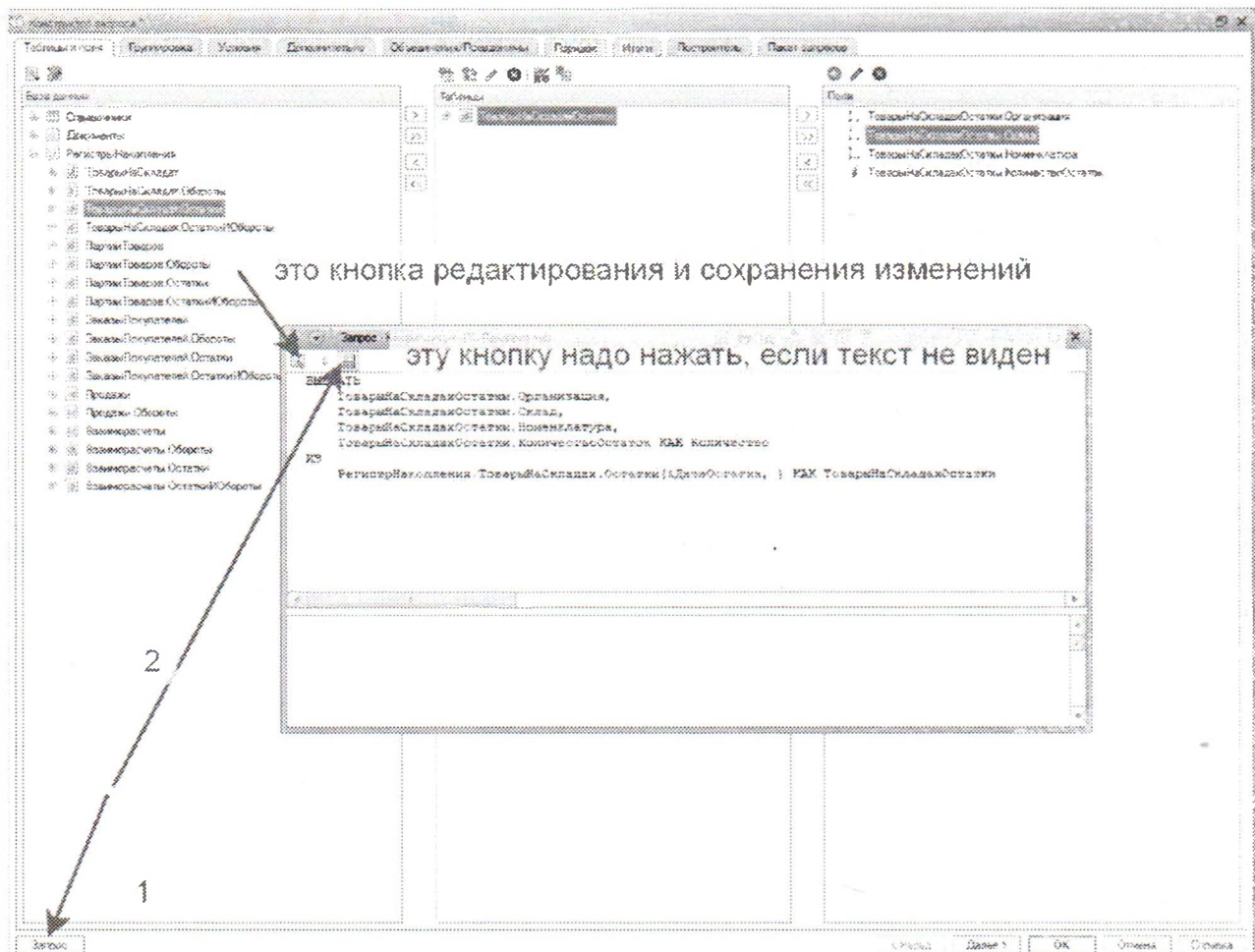


Рисунок 123

Выделяем текст запроса и копируем его (сочетание клавиш ctrl+C). Далее переходим в конфигуратор, позиционируемся в строке присвоения текста запросу (Рисунок 124).

```
Запрос.Текст = '';
```

Рисунок 124

Вызываем из контекстного меню «Конструктор запроса». Затем в том же порядке нажимаем кнопку «Запрос», кнопку редактирования, вставляем наш текст из буфера, еще раз нажимаем кнопку редактирования, соглашаемся с внесенными изменениями и нажимаем «Ок». Запрос в программном коде оформлен (Рисунок 125-126).

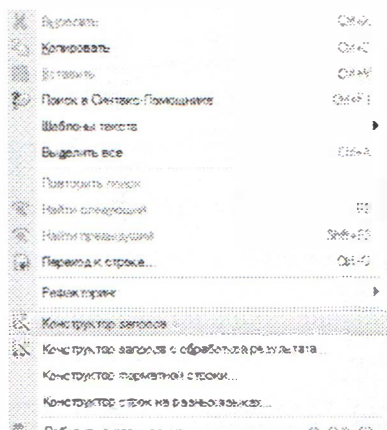


Рисунок 125

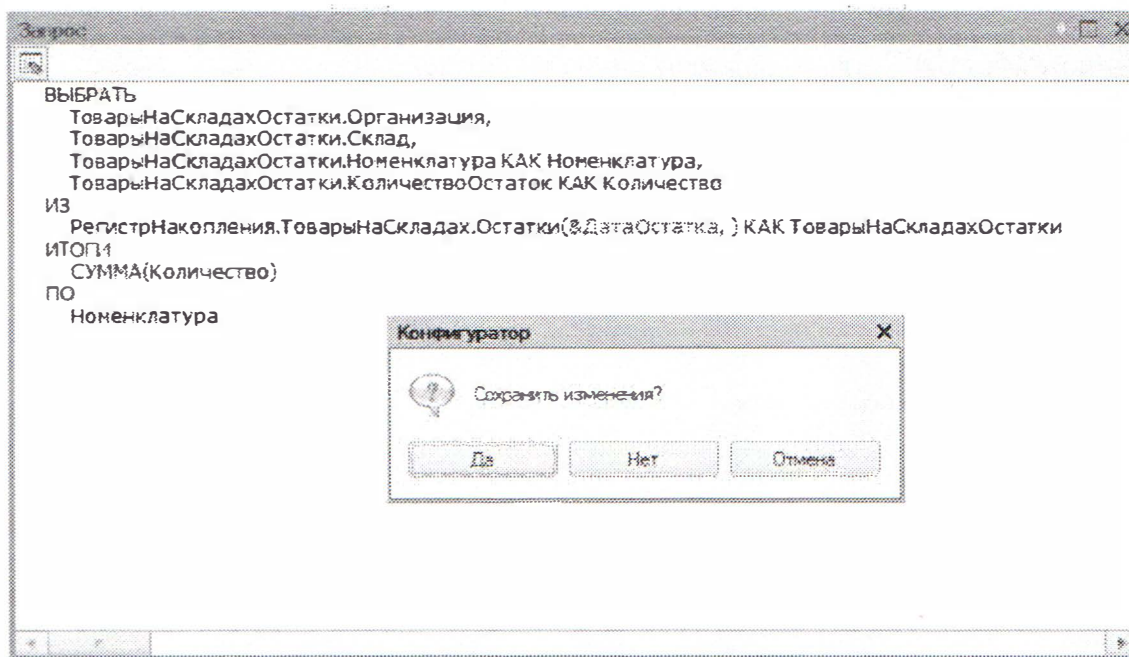


Рисунок 126

Нам осталось установить параметр «ДатаОстатка» для нашего запроса. В тексте уже установлены параметры «НачПериода» и «КонПериода». Устанавливаются они значениями одноименных реквизитов отчета. Обратите внимание, что на форму выведен только реквизит «КонПериода», его подпись на форме – «Дата актуальности» (Рисунок 127).

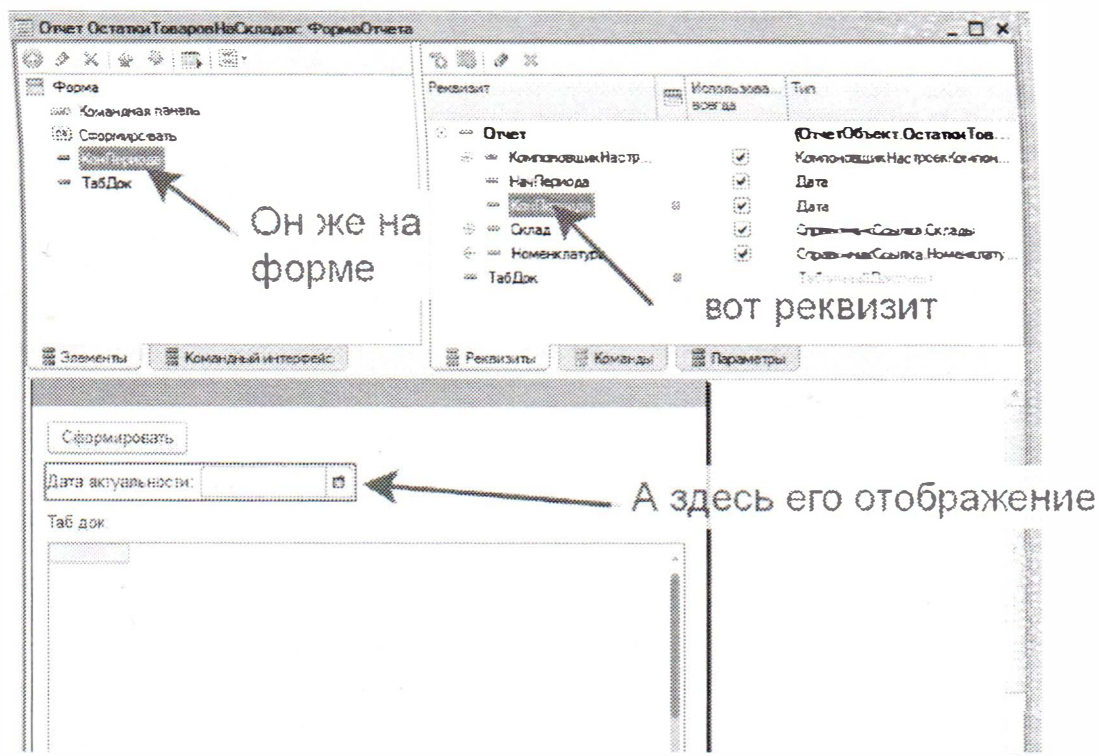


Рисунок 127

Немного изменим наш программный код в части установки параметров запроса. Вместо строки «Запрос.УстановитьПараметр("КонПериода".КонПериода);», напишем «Запрос.УстановитьПараметр("ДатаОстатка".КонПериода);», а строчку «Запрос.УстановитьПараметр("НачПериода".НачПериода);» – закомментируем.

В итоге, получим вот такой текст процедуры (Рисунок 128).

```

&НаСервереБезКонтекста
Процедура СформироватьНаСервере(ТабДок, НачПериода, КонПериода, Склад, Номенклатура)
    Макет = Отчеты.ОстаткиТоваровНаСкладах.ПолучитьМакет("Макет");
    ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
    ОбластьОрганизация = Макет.ПолучитьОбласть("Организация");
    ОбластьСклад = Макет.ПолучитьОбласть("Склад");
    ОбластьНоменклатура = Макет.ПолучитьОбласть("Строка");

    ТабДок.Очистить();
    ТабДок.Вывести(ОбластьШапка);

    Запрос = Новый Запрос;
    //Запрос.УстановитьПараметр("НачПериода", НачПериода);
    Запрос.УстановитьПараметр("ДатаОстатка", КонПериода);
    Запрос.Текст = "ВЫБРАТЬ
        | ТоварыНаСкладахОстатки.Организация,
        | ТоварыНаСкладахОстатки.Склад,
        | ТоварыНаСкладахОстатки.Номенклатура КАК Номенклатура,
        | ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
    |ИЗ
        | РегистрНакопления.ТоварыНаСкладах.Остатки(&ДатаОстатка, ) КАК ТоварыНаСкладахОстатки
    |ИТОГИ
        | СУММА(Количество) |
    |ПО
        | Номенклатура";
КонецПроцедуры

```

Рисунок 128

2.6 Выборка с итогами в отладчике.

В этом занятии мы научимся работать с выборками, содержащими итоги во встроеном языке. Из запроса, в котором будут содержаться итоги, мы получим выборку, но не просто выборку, а итоговую выборку. В свою очередь, детальную выборку мы будем получать из итоговой выборки. Такой подход позволяет безошибочно выбирать итоги и выводить их в отчет, то есть мы всегда будем четко понимать, какие итоги получаем при обходе выборки.

На рисунке 129 показаны три способа обхода выборки: линейный обход, иерархический обход и обход по группировкам. Данные, попадающие в выборку, выделены цветом. Нас интересует, прежде всего, линейный обход и обход по группировкам. При линейном обходе выборки мы последовательно получаем все записи из результата запроса, при иерархическом обходе – только записи результата запроса, находящиеся на одном уровне иерархии, при обходе по группировкам будут получены только родительские записи, являющиеся групповыми итогами.

При расчете итогов образуется некая иерархическая структура, поскольку записи с одинаковым значением поля (полей), для которых рассчитываются итоги, в результате запроса собираются (группируются) вместе и достраиваются итоговой строкой. Эта итоговая строка является, по отношению к ним, родительской, то есть стоящей на более высоком уровне иерархии.

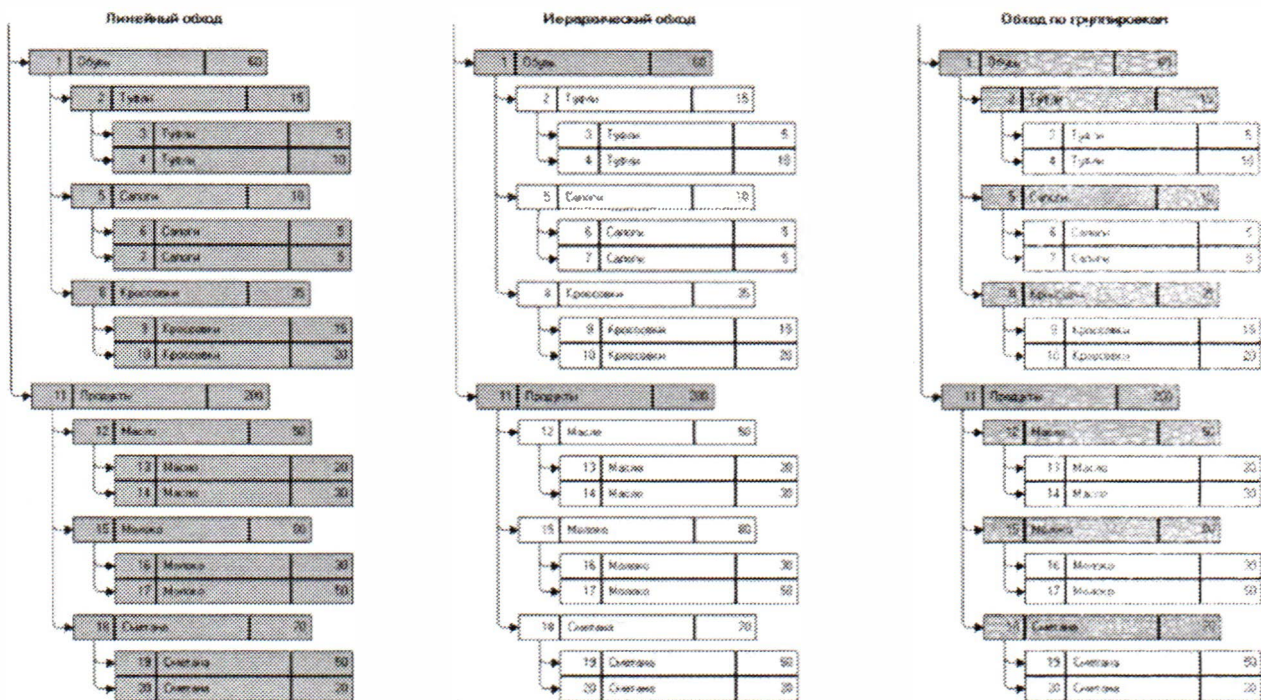


Рисунок 129

Для дальнейшей работы нам потребуется несколько изменить запрос: добавить группировочные поля «Склад» и «Организация» в итоги, а поле «Номенклатура» из итогов убрать (Рисунок 130).

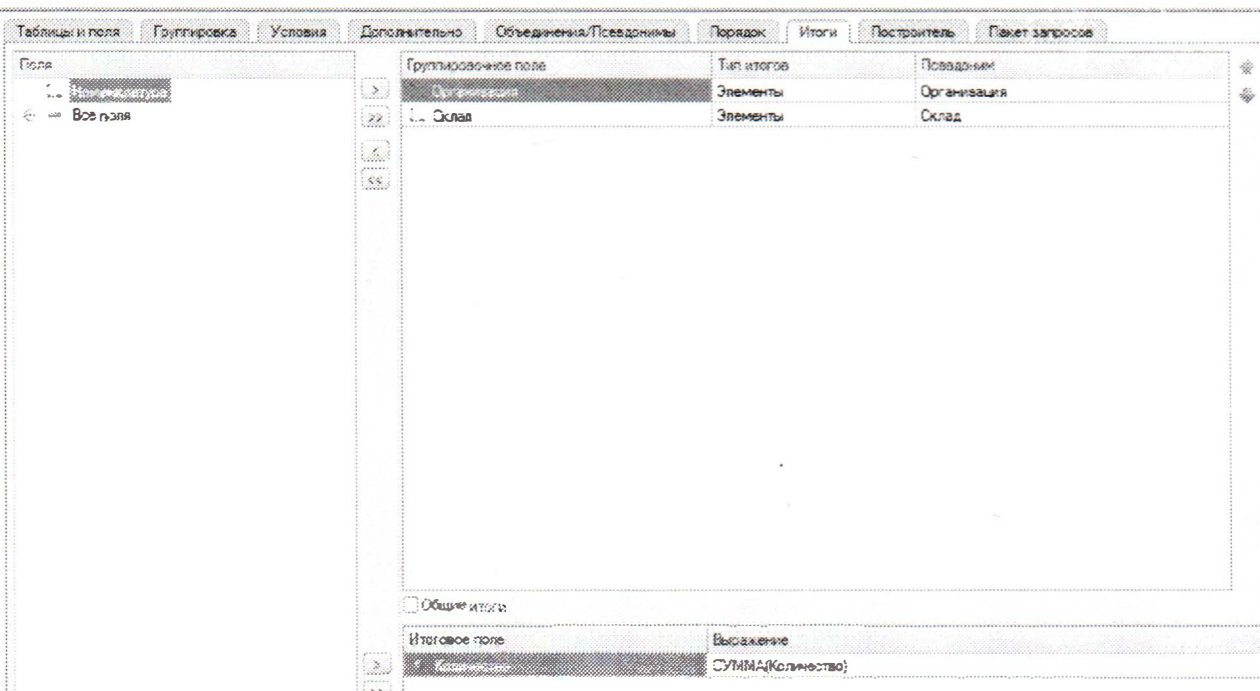


Рисунок 130

Далее нам необходимо выполнить обход по группировкам. Для начала мы должны получить итоговую выборку по организации, назовем ее ВыборкаОрганизация, затем итоговую выборку по складу, назовем ее ВыборкаСклад, а в конце детальные записи по

номенклатуре, назовем их Выборка. В программном коде это будет выглядеть следующим образом (Рисунок 131).

```
РезультатЗапроса = Запрос.Выполнить();
ВыборкаОрганизация = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
Пока ВыборкаОрганизация.Следующий() Цикл
    ВыборкаСклад = ВыборкаОрганизация.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
    Пока ВыборкаСклад.Следующий() Цикл

    Выборка = ВыборкаСклад.Выбрать();
    Пока Выборка.Следующий() Цикл
КонецЦикла;
КонецЦикла;
КонецЦикла;
```

Рисунок 131

Немного поясним этот код.

Внутри циклов мы пока никаких действий не выполняем, а только получаем следующий элемент выборки. Выборку самого верхнего уровня мы получаем из результата запроса при помощи метода Выбрать(), передавая ему, в качестве параметра, значение системного перечисления «ОбходРезультатаЗапроса». Это системное перечисление (для справки скажу, что системные перечисления предназначены для определения некоторого набора предопределенных значений), может принимать значения: Прямой, ПоГруппировкам, ПоГруппировкамСИерархией. Мы передаем значение ПоГруппировкам. Таким образом, мы получаем выборку верхнего уровня с итогами по организации. Затем мы обходим эту выборку в цикле, а внутри этого цикла получаем выборку следующего уровня с итогами по складу:

```
ВыборкаСклад =
ВыборкаОрганизация.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
```

То есть, выборку самого верхнего уровня мы получаем из результата запроса, а выборки всех последующих уровней – из родительских выборок. Поскольку выборка самого нижнего уровня содержит в себе детальные записи, то есть не содержит итогов, то при ее получении мы опускаем параметр «Тип обхода», так как по умолчанию он и так устанавливается как «Прямой».

Теперь посмотрим в отладчике, какие же значения получаются в каждой выборке и как все это работает.

Для этого сохраним изменения в конфигурации и запустим отладку. Откроем наш «Отчет по остаткам». Установим точку останова в начало цикла обхода выборки по организации. Нажмем на кнопку отчета «Сформировать» (Рисунок 132).

```

РезультатЗапроса := Запрос.Выполнить();

ВыборкаОрганизация := РезультатЗапроса.Выбрать{ОбходРезультатаЗапроса.ПоГруппировкам};

Пока ВыборкаОрганизация.Следующий() Цикл

    ВыборкаСклад := ВыборкаОрганизация.Выбрать{ОбходРезультатаЗапроса.ПоГруппировкам};

    Пока ВыборкаСклад.Следующий() Цикл

        Выборка := ВыборкаСклад.Выбрать();

        Пока Выборка.Следующий() Цикл

    КонечЦикла;
КонечЦикла;
КонечЦикла;

```

Рисунок 132

Нажмем кнопку F11 и попадем внутрь цикла. Теперь мы можем посмотреть на значение выборки по организации. Для этого выделим переменную «ВыборкаОрганизация» и нажмем сочетание клавиш «Shift+F9» (Рисунок 133).

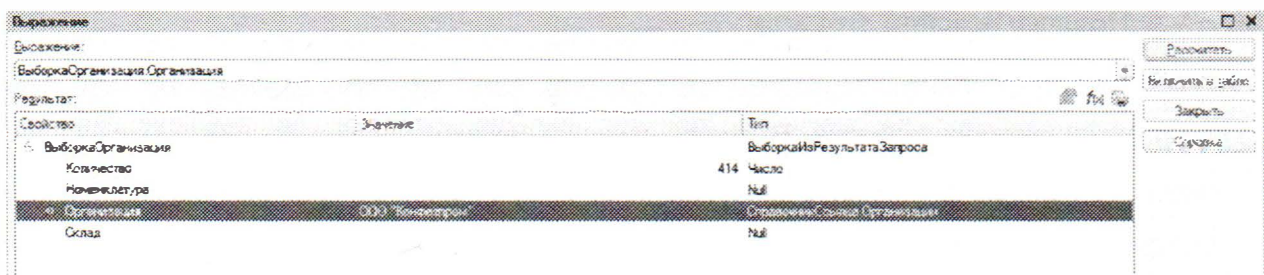


Рисунок 133

Мы видим, что в организации ООО «Конфетпром» 414 единиц товаров. Значения полей «Номенклатура» и «Склад» равны Null, так как эти поля не являются ни групповыми, ни суммируемыми для данного уровня выборки.

Идем далее и получаем выборку с итогами по складу. Заходим в цикл, обходящий итоги по складу и просматриваем значение переменной «ВыборкаСклад» (Рисунок 134).

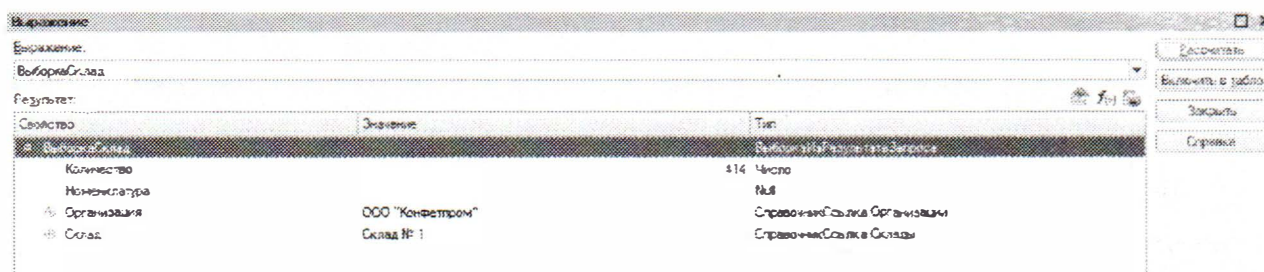


Рисунок 134

Итак, мы видим, что в организации ООО «Конфетпром» на складе «Склад №1» имеется 414 единиц каких-то товаров. Поле номенклатура остается равно Null.

Наконец, двигаемся далее и получаем выборку с детальными записями. Заходим внутрь цикла обхода детальных записей, и там просматриваем значение выборки (Рисунок 135).

Рисунок 135

Здесь мы видим, что все поля выборки заполнены, так как это детальные записи. Таким образом, при обходе этой выборки, мы получаем расшифровку по каждой номенклатурной позиции из итоговых 414 единиц, которые имеются на складе в организации.

2.7 Отчёт с итогами.

В этой части выведем данные нашего отчета в табличный документ.

Для этого нужно совсем немного дописать наш программный код. Суть доработки заключается в том, что мы заполним параметры соответствующих областей шаблона и выведем эти области в результирующий табличный документ.

Для начала выведем итоги по организации. Для этого возьмем область итогов по организации, которая у нас обозначена как «ОбластьОрганизация», заполним ее параметры и, затем, выведем в табличный документ. Для этого в цикле обхода выборки по организации добавим 2 строки, отображенные на рисунке 136.

```
Пока ВыборкаОрганизация.Следующий() Цикл
  ОбластьОрганизация.Параметры.Заполнить(ВыборкаОрганизация);
  ТабДок.Вывести(ОбластьОрганизация);|
```

Рисунок 136

Первая из добавленных строк заполняет параметры области, это возможно благодаря тому, что наименование полей выборки совпадает с наименованием параметров области (Рисунок 137).

А: Отчет ОстаткиТоваровНаСкладах: Макет											
	1	2	3	4	5	6	7	8	9	10	
Шапка	2	Организация / склад / номенклатура							Количество		
Организация	5	<Организация>					<Количество>				
Склад	7	<Склад>				<Количество>					
Строка	9	<номенклатура>				<Количество>					
	10										
	11										

это и есть параметры области "Организация"

Рисунок 137

То же самое мы проделываем внутри циклов обхода выборок по складу и детальным записям, подставляя соответствующие области и выборки для заполнения (Рисунок 138).

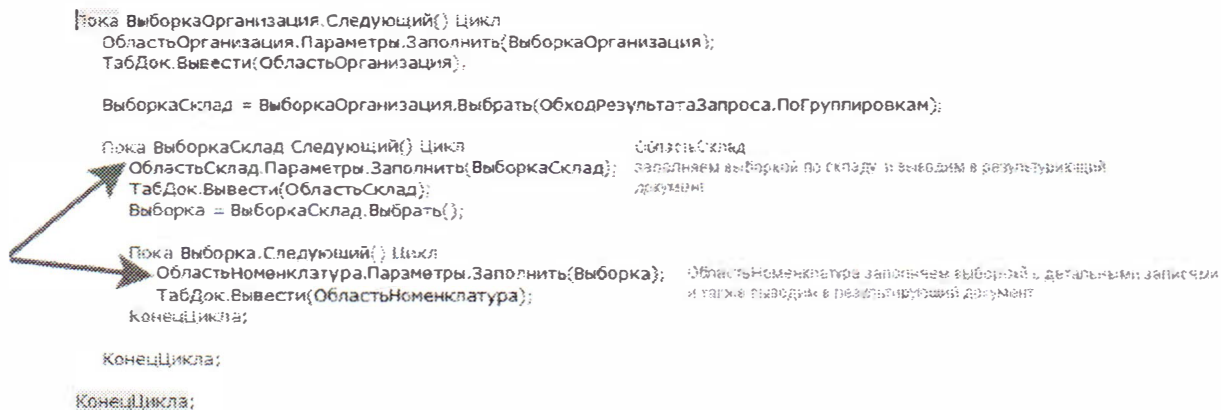


Рисунок 138

Таким образом, макет играет роль некой заготовки, которую можно по частям (по областям) выводить в табличный документ. При этом в макете могут быть заданы параметры, которые можно заполнять произвольными значениями.

Обновим конфигурацию, зайдём в режим «1С:Предприятие» и сформируем отчет.

Вот таким нехитрым способом мы сформировали табличный документ, который отображает наш отчет (Рисунок 139).



Остатки товаров на складах

Сформировать

Дата актуальности: 10.07.2014

Таб док

Организация / склад / номенклатура	Количество
ООО "Конфетпром"	414
Склад № 1	414
Вентилятор WINATONE ALPINE 160вт, напольный .	90
Вентилятор JIPONIC (Тайв.).	90
Вентилятор настольный	30
Вентилятор оконный	10
Вентилятор ОРБИТА, STERLING, ЯП.	150
Кондиционер ELEKTA	44
ООО "Торговый дом"	926
Склад № 1	620
СТИНОЛ 101	22
Вентилятор WINATONE ALPINE 160вт, напольный .	22
Вентилятор JIPONIC (Тайв.).	80
Вентилятор настольный	189
Вентилятор оконный	170
Вентилятор ОРБИТА, STERLING, ЯП.	30
Телевизор "JVC"	15
Кондиционер ELEKTA	67
Телевизор "SHARP"	25
Склад № 2	301
Вентилятор WINATONE ALPINE 160вт, напольный .	94
Вентилятор JIPONIC (Тайв.).	32
Вентилятор настольный	40
Вентилятор оконный	15
Вентилятор ОРБИТА, STERLING, ЯП.	100
Кондиционер ELEKTA	20
Склад № 3	5
Вентилятор настольный	5

Рисунок 139

2.8 Общие итоги.

Продолжая тему итогов в запросе, мы должны обратить внимание на общие итоги в запросе, которые выводят значение агрегатной функции, применительно ко всем строкам, попавшим в выборку (ко всем детальным записям). В нашем случае агрегатная функция — это сумма, поэтому общие итоги будут суммой поля «Количество» всех строк детальных записей.

Как получить общие итоги? Да очень просто. Необходимо выполнить 2 действия в конструкторе запроса на закладке «Итоги» (Рисунок 140):

1. Указать хотя бы одно итоговое поле.
2. Установить флажок «Общие итоги».

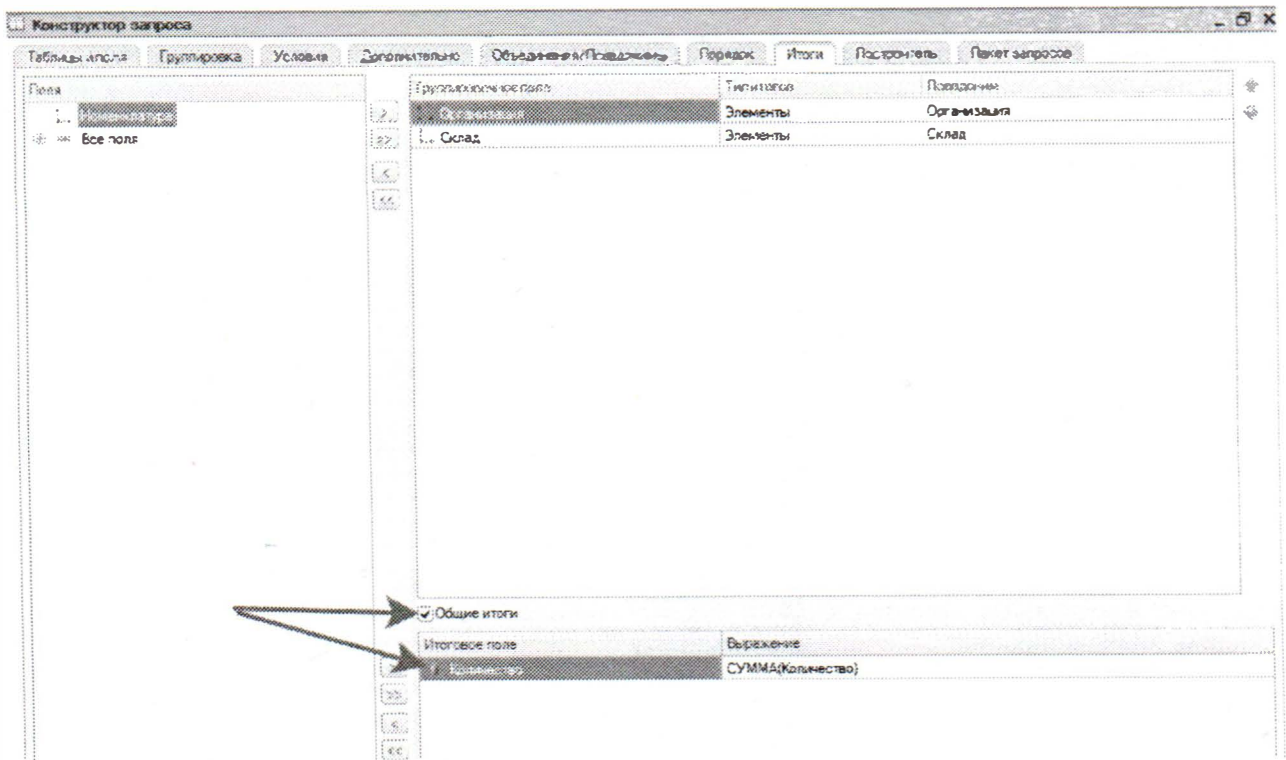


Рисунок 140

Сделаем это в нашем запросе и посмотрим, как изменился текст запроса (Рисунок 141).

```

ВЫБРАТЬ
    ТоварыНаСкладахОстатки.Организация КАК Организация,
    ТоварыНаСкладахОстатки.Склад КАК Склад,
    ТоварыНаСкладахОстатки.Номенклатура,
    ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
ИЗ
    РегистрНакопления.ТоварыНаСкладах.Остатки(;&ДатаОстатка, ) КАК ТоварыНаСкладахОстатки
ИТОГИ
    СУММА (Количество)
ПО
    ОБЩИЕ,
    Организация,
    Склад
  
```

Рисунок 141

Видим, что в секцию итогов добавилась строка с ключевым словом «Общие». А как изменился сам результат запроса? (Рисунок 142)

7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	000 "Конфетпром"	Склад № 1	Вентилятор оконный																				
	000 "Конфетпром"	Склад № 1	Вентилятор СРБИТА.STERLING,ЯП.																				
	000 "Конфетпром"	Склад № 1	КондиционерELEKTA																				
	000 "Торговый дом"	Склад № 1																					
	000 "Торговый дом"	Склад № 1	СТИНОЛ 151																				
	000 "Торговый дом"	Склад № 1	Вентилятор ВIKATONE ALPHE 160вт, напольный.																				
	000 "Торговый дом"	Склад № 1	Вентилятор JPOHC (Tайе).																				
	000 "Торговый дом"	Склад № 1	Вентилятор настольный																				
	000 "Торговый дом"	Склад № 1	Вентилятор оконный																				
	000 "Торговый дом"	Склад № 1	Вентилятор СРБИТА.STERLING,ЯП.																				
	000 "Торговый дом"	Склад № 1	Телевизор "JVC"																				
	000 "Торговый дом"	Склад № 1	Кондиционер ELEKTA																				
	000 "Торговый дом"	Склад № 1	Телевизор "SHARP"																				
	000 "Торговый дом"	Склад № 2																					
	000 "Торговый дом"	Склад № 2	Вентилятор ВIKATONE ALPHE 160вт, напольный.																				
	000 "Торговый дом"	Склад № 2	Вентилятор JPOHC (Tайе).																				
	000 "Торговый дом"	Склад № 2	Вентилятор настольный																				
	000 "Торговый дом"	Склад № 2	Вентилятор оконный																				
	000 "Торговый дом"	Склад № 2	Вентилятор СРБИТА.STERLING,ЯП.																				
	000 "Торговый дом"	Склад № 2	Кондиционер ELEKTA																				
	000 "Торговый дом"	Склад № 3																					
	000 "Торговый дом"	Склад № 3	Вентилятор настольный																				
	Итого																						

Рисунок 142

Появилась еще одна строка с общим итогом по количеству.

Необходимо отметить еще одну важную деталь – при программной обработке выборки, выборка самого верхнего уровня будет с общими итогами.

То есть, в строке:

«ВыборкаОрганизация =

РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);»

мы получим выборку с общими итогами, а не с итогами по организации. Но мы можем напрямую указать, какого уровня выборку мы хотим получить, передав вторым параметром в метод «Выбрать» строку с наименованием группировочного поля (Рисунок 143).

РезультатЗапроса = Запрос.Выполнить();

ВыборкаОрганизация = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам| "Организация");

Рисунок 143

Таким образом, мы явно указали, какую группировку хотим получить, и, в данном случае, несмотря на наличие общих итогов, наш программный код будет выполнен, как и ранее.

2.9 Виртуальная таблица. Регистр накопления обороты.

В этой части мы рассмотрим виртуальную таблицу оборотов оборотного регистра накопления. Напомним, регистры накопления бывают двух видов: «остатки» и «обороты».

Регистры накопления с видом «остатки» хранят в себе остатки, по ним могут быть приходные и расходные движения. Регистры накопления вида «обороты» не хранят в себе остатков, они отражают только объем чего-либо (например, сколько всего было продано товаров). Естественно, обороты имеют смысл только за какой-то период, в отличие от

остатков, которые получаются на определенную дату. У оборотного регистра существует только одна виртуальная таблица обороты с суффиксом «обороты». Исходя из вышесказанного, логично предположить, что параметрами такой виртуальной таблицы являются: начало периода и конец периода (то есть период, по которому требуется получить обороты). Кроме того, параметром виртуальной таблицы оборотов является периодичность, с которой требуется выводить обороты. Этот параметр задает минимальный период, в котором они рассматриваются. Если мы выберем месяц, то обороты будут выведены в пределах месяца (помесечно), в рамках заданного периода (начала периода и окончания периода).

Давайте рассмотрим это на конкретном примере. Для этого в режиме предприятия запустим консоль запросов. В ней сформируем новый запрос. Обратимся к виртуальной таблице «ПродажиОбороты». Это виртуальная таблица оборотов оборотного регистра накопления «Продажи». Таблица состоит из измерений: «Поставщик», «Организация», «Склад», «Номенклатура» и ресурсов: «КоличествоОборот», «СуммаОборот». Выберем из нее данные за весь период и все поля, периодичность установим «День».

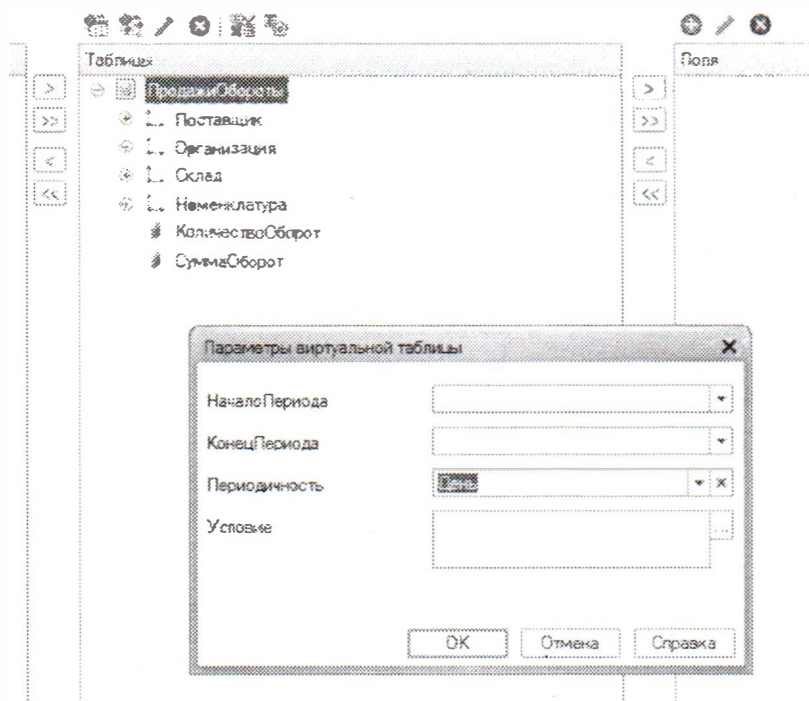


Рисунок 144

После установки периодичности у виртуальной таблицы появилось еще одно поле «Период». Теперь выберем все поля (Рисунок 145).

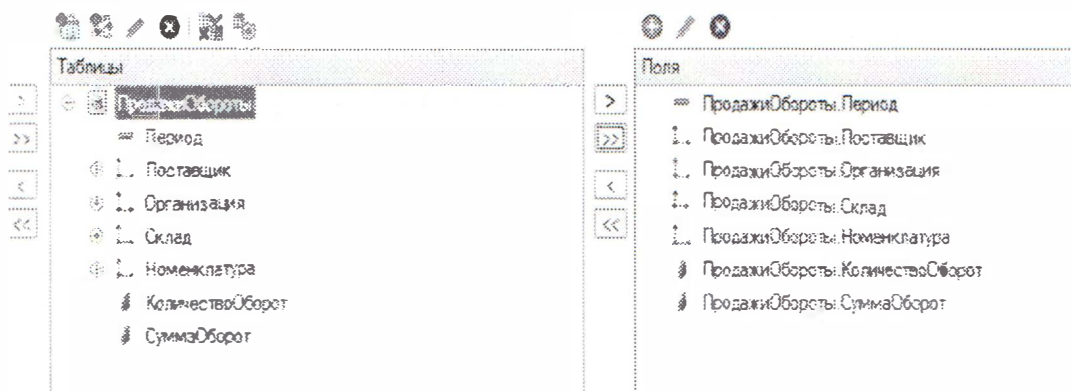


Рисунок 145

Выполним запрос и посмотрим на результат (Рисунок 146).

Результат запроса

1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Период		Поставщик	Организация		Склад		Номенклатура		КоличествоОборот		СуммаОборот	
2	22.01.2014 0:00:00			ООО "Торговый дом"		Склад № 1		Вентилятор оконный		10		5 200,00	
3	30.01.2014 0:00:00			ООО "Торговый дом"		Склад № 1		Телевизор "JVC"		5		90 000,00	
4	30.01.2014 0:00:00			ООО "Торговый дом"		Склад № 1		Телевизор "SHARP"		10		290 000,00	
5													
6													
7													

Рисунок 146

Мы видим все продажи, которые были сделаны. Период заполнен началом дня, так как мы установили периодичность «День».

Теперь установим периодичность «Неделя» и вновь выполним запрос. В этот раз период заполнен началом недели. То есть, точность оборотов по дате «округляется» до начала недели. Для подтверждения данного тезиса посмотрим документы продажи. У нас их два (Рисунок 147).

← → ☆ Реализация товаров и услуг

Создать Панель (Ctrl+F) × Q - Еще ▾

Дата	Номер	Организация	Склад	Контрагент	Сумма документа	Заказ
22.01.2014 12:00:00	000000001	ООО "Торговый до ...	Склад № 1	Магазин на тагыш...	5 200,00	Заказ
30.01.2014 12:00:00	000000002	ООО "Торговый до ...	Склад № 1	Магазин на тагыш...	380 000,00	Заказ

Рисунок 147

Реализация от 22.01.2014 «округляется» до начала недели – 20.01.2014, а реализация от 30.01.2014 до 27.01.2014г.

Результат запроса

1	2	3	4	5	6	7	8	9
	Период	Поставщик	Организация	Склад	Номенклатура	КоличествоОборот	СуммаОборот	
2	20.01.2014 0:00:00		ООО "Торговый дом"	Склад № 1	Вентилятор оконный	10	5 200,00	
3	27.01.2014 0:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "JVC"	5	90 000,00	
4	27.01.2014 0:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "SHARP"	10	290 000,00	
5								

Рисунок 148

Номенклатура, по которой не было оборотов, не выводится в результат вообще. В результат попадают только те позиции, по которым были обороты.

Теперь установим периодичность «Регистратор». Регистратор – это документ, который послужил основанием для формирования данного движения. То есть, поле «Период» будет заполнен датой документа регистратора. Кроме того, появится дополнительное поле «Регистратор», которое будет содержать ссылку на этот документ. Давайте изменим наш запрос, установив периодичность «Регистратор» и добавив поле «Регистратор» в выходные поля запроса (Рисунок 149).

Время выполнения: - - - - - Время вывода: - - - - - Количество: 3

Результат запроса

1	2	3	4	5	6	7	8	9	
Период	Поставщик	Организация	Склад	Номенклатура	Количество	Оборот	Сумма	Оборот	Регистратор
22.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Вентилятор оконный	10	5 200,00	Реализация товаров и услуг	00000001	от 22.01.2014 12:00:00
30.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "JVC"	5	90 000,00	Реализация товаров и услуг	00000002	от 30.01.2014 12:00:00
30.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "SHARP"	10	290 000,00	Реализация товаров и услуг	00000003	от 30.01.2014 12:00:00

Рисунок 149

Как видим, ссылка на документ регистратор выводится в результат, а поле «Период» заполнено датой документа регистратора. Соответственно, мы получили обороты по каждому документу.

Если же мы установим периодичность «Запись», то в выходные поля таблицы добавится поле «НомерСтроки» и нами будут получены обороты по каждой номенклатуре в каждом документе (Рисунок 150).

Результат запроса

1	2	3	4	5	6	7	8	9	10	
Период	Поставщик	Организация	Склад	Номенклатура	Количество	Оборот	Сумма	Оборот	Регистратор	НомерСтроки
22.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Вентилятор оконный	10	5 200,00	Реализация товаров и услуг	00000001	от 22.01.2014 12:00:00	1
30.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "JVC"	5	90 000,00	Реализация товаров и услуг	00000002	от 30.01.2014 12:00:00	1
30.01.2014 12:00:00		ООО "Торговый дом"	Склад № 1	Телевизор "SHARP"	10	290 000,00	Реализация товаров и услуг	00000003	от 30.01.2014 12:00:00	2

Рисунок 150

Итак, в результате мы видим периодичности: Секунда, Год, Минута, Час, Полугодие, Декада, Квартал. Рассматривать каждое по-отдельности не будем, скажем лишь то, что работают они аналогично. Установим периодичность «Авто» и увидим, что в полях виртуальной таблицы появились все периодичности (Рисунок 151).

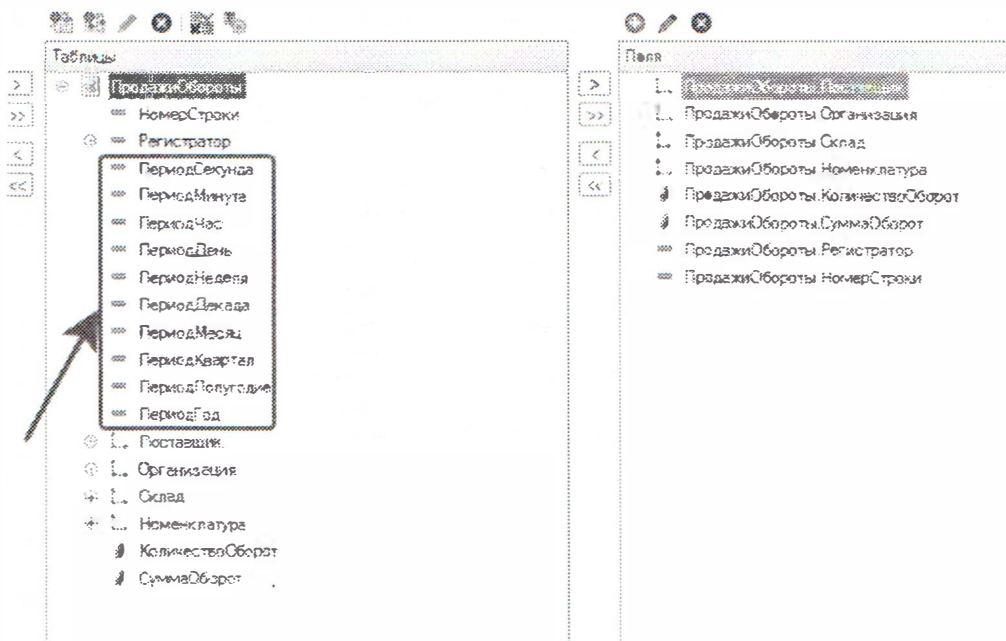


Рисунок 151

Таким образом, при необходимости, можно одновременно получить любую из возможных периодичностей.

В параметрах виртуальной таблицы мы также можем задать условия, но только на измерения (Рисунок 152).

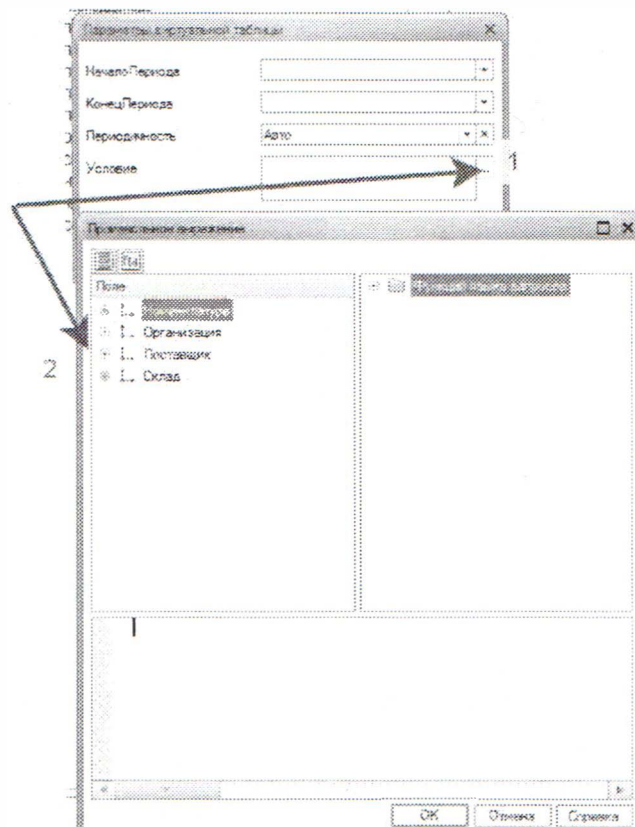


Рисунок 152

Если в регистре будут какие-то реквизиты (реквизиты в регистре накопления могут как-то дополнительно характеризовать движения), то наложить условия на реквизиты в виртуальной таблице мы не сможем.

2.10 Виртуальная таблица. Регистр накопления остатки.

Рассмотрим регистр накопления вида «остатки». С ним мы уже сталкивались в предыдущих занятиях. В отличие от оборотного регистра, он имеет две виртуальные таблицы: «Остатки» и «ОстаткиИОбороты». Виртуальная таблица «ОстаткиИОбороты» во многом схожа с рассмотренной в девятой части таблицей «Обороты», но в ней, помимо прочего, присутствует информация об остатках на начало периода и на конец периода. Рассмотрим мы ее чуть позже. Таблица «Остатки» имеет параметр «Период» (дату, на которую требуется получить остатки) и условие по измерениям, так же, как и в оборотной таблице. Если оставить поле «Период» пустым, то остатки будут получены на текущую дату.

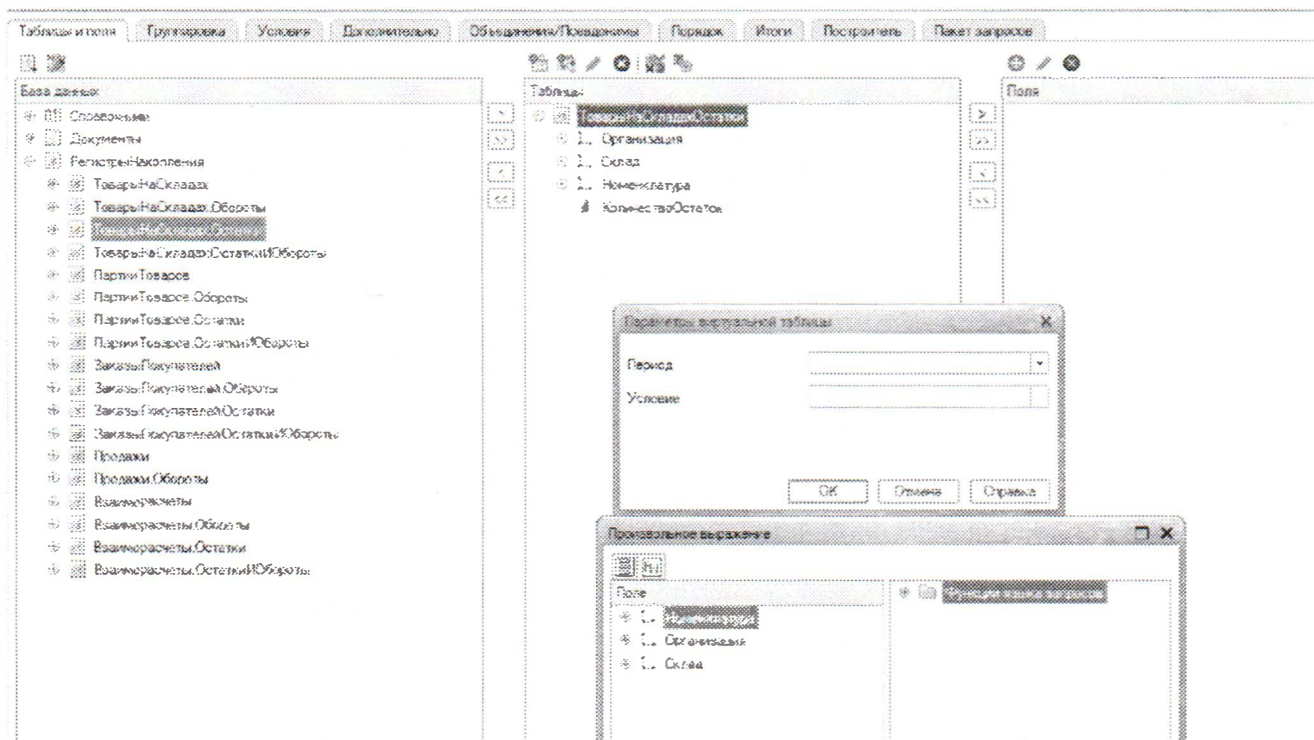


Рисунок 153

2.11 Виртуальная таблица. Регистр накопления Остатки и обороты.

Как упоминалось ранее, у регистра накопления вида остатки есть виртуальная таблица «ОстаткиИОбороты». Она содержит информацию не только по оборотам, но и по приходу, расходу, остаткам на начало периода и остаткам на конец периода. Среди параметров, аналогично таблице «Обороты» оборотного регистра, имеет начало периода, конец периода, периодичность и условие на измерения. Периодичность такая же, как и в регистре «Обороты» оборотного регистра, который мы рассмотрели в 9-й части. Если мы, для примера, выберем периодичность месяц, то получим остатки на начало месяца, на конец

месяца, а также приход и расход за месяц. Для примера возьмем таблицу «ТоварыНаСкладахОстаткиИОбороты», установим периодичность «День» и посмотрим на результат запроса (Рисунок 154).

Время выполнения: - Время вывода: - Количество: 29

Результат запроса

1	2	3	4	5	6	7	8	9	10
Период	Организация	Склад	Номенклатура	КоличествоНача	КоличествоКонеч	КоличествоОборот	КоличествоКопи		
0: 01.2014 0:00:00		Склад № 1	Вентилятор ВЕНАТОНЕ ALPINE 150вт. настольный.		90		90		
3: 01.2014 0:00:00		Склад № 1	Вентилятор ВЕНАТОНЕ ALPINE 150вт. настольный.	90	112	22	22		
0: 01.2014 0:00:00		Склад № 1	Вентилятор JIPONIC (Тайпе).		90	90	90		
3: 01.2014 0:00:00		Склад № 1	Вентилятор JIPONIC (Тайпе).	90	170	80	80		
0: 01.2014 0:00:00		Склад № 1	Вентилятор настольный		30	30	30		
3: 01.2014 0:00:00		Склад № 1	Вентилятор настольный	30	74	44	44		
0: 01.2014 0:00:00		Склад № 1	Вентилятор оконный		10	10	10		
3: 01.2014 0:00:00		Склад № 1	Вентилятор оконный	10	70	60	60		
0: 01.2014 0:00:00		Склад № 1	Вентилятор ОРБИТА STERLING,ЯП.		150	150	150		
3: 01.2014 0:00:00		Склад № 1	Вентилятор ОРБИТА STERLING,ЯП.	150	180	30	30		
0: 01.2014 0:00:00		Склад № 1	Кондиционер ELEKTA		44	44	44		
3: 01.2014 0:00:00		Склад № 1	Кондиционер ELEKTA	44	84	40	40		
15: 01.2014 0:00:00		Склад № 2	Вентилятор ВЕНАТОНЕ ALPINE 150вт. настольный.		94	94	94		
15: 01.2014 0:00:00		Склад № 2	Вентилятор JIPONIC (Тайпе).		32	32	32		
15: 01.2014 0:00:00		Склад № 2	Вентилятор настольный		40	40	40		
15: 01.2014 0:00:00		Склад № 2	Вентилятор оконный		15	15	15		
15: 01.2014 0:00:00		Склад № 2	Вентилятор ОРБИТА STERLING,ЯП.		100	100	100		
15: 01.2014 0:00:00		Склад № 2	Кондиционер ELEKTA		20	20	20		
22: 01.2014 0:00:00	ООО	Склад № 1	СТИНОЛ 101		22	22	22		

Рисунок 154

Мы видим, что остатки выведены за каждый день по тем позициям, по которым были остатки или движения за период. Здесь необходимо сделать важное замечание: никогда не используйте таблицу «ОстаткиИОбороты», если вам нужны только остатки или только обороты. Это продиктовано соображениями оптимизации, так как таблица «ОстаткиИОбороты» требует больше ресурсов для работы с ней. Неоправданное обращение к этой таблице способно заметно замедлить работу системы, чего, конечно же, нужно стараться избегать при разработке запросов.

2.12 Задача на соединение таблиц.

Часто в результате запроса требуется вывести данные из разных таблиц, связав их по значению некоторого поля. Другими словами, какие-то поля вывести из одной таблицы, какие-то – из другой. Например, можно вывести для одного и того же товара количество его поступлений и продаж. В этом случае используются несколько источников данных, которые перечисляются после ключевого слова «ИЗ». В качестве источников запроса могут выступать реальные и виртуальные таблицы, а также вложенные запросы, но пока, для упрощения примера, мы будем рассматривать ситуацию, когда в качестве источников запроса выступают две таблицы базы данных. Суть примера от этого не изменится.

Исходные таблицы запроса обычно связываются (соединяются) между собой по некоторому условию – условию связи. Поле, по которому производится связь, обычно имеет

ссылочный тип. При соединении данных из исходных таблиц запроса для каждой записи из этих таблиц проверяется условие равенства полей связи. Условие связи источников запроса задается в предложении «ИЗ», после ключевого слова «ПО». Например, «ПО Товары.Ссылка = Цены.Товар». Таблицы могут соединяться внутренним, левым внешним, правым внешним или полным внешним соединением.

В нашем примере, мы возьмем данные из табличной части документа и соединим их с таблицей остатков регистра накопления «Товары на складах». Так мы получим текущие остатки для каждой позиции номенклатуры из документа. Для чего это может понадобиться? Прежде всего, для контроля остатков при проведении документа.

Итак, берем, в качестве первой таблицы, табличную часть «Товары» документа «Реализация товаров и услуг». Для того чтобы не выбрать табличные части всех документов, мы зададим условие на поле «Ссылка». Из этой таблицы мы выберем поля «Номенклатура» и «Количество» (Рисунок 155-156).

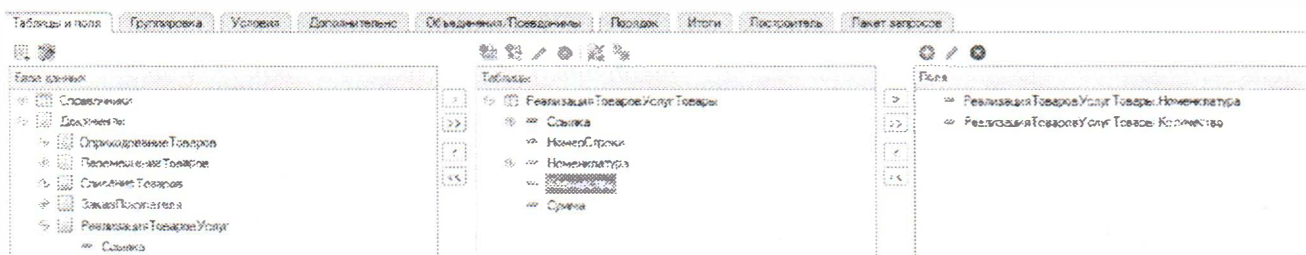


Рисунок 155

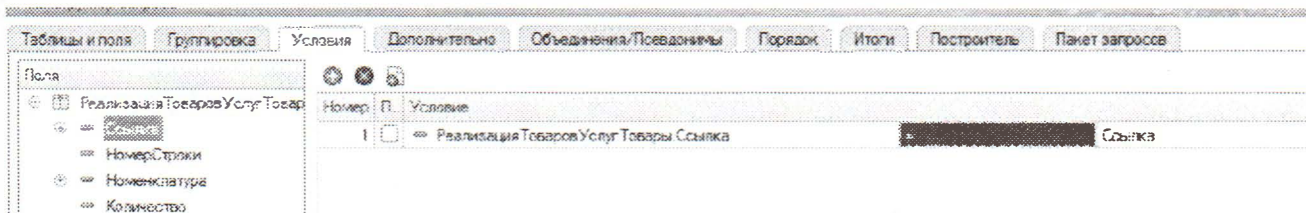


Рисунок 156

В качестве второго источника данных, выберем виртуальную таблицу остатки регистра накопления «Товары на складах». Для установки даты получения остатков зададим параметр «период», а также наложим условие на измерения «Организация» и «Склад». Если бы мы работали с запросом во встроенном языке, то мы бы взяли эти параметры из документа, но поскольку сейчас мы работаем в консоли запросов, эти параметры придется устанавливать вручную. Конечно же, параметры «Организация» и «Склад» должны быть установлены такими же, как в документе (Рисунок 157).

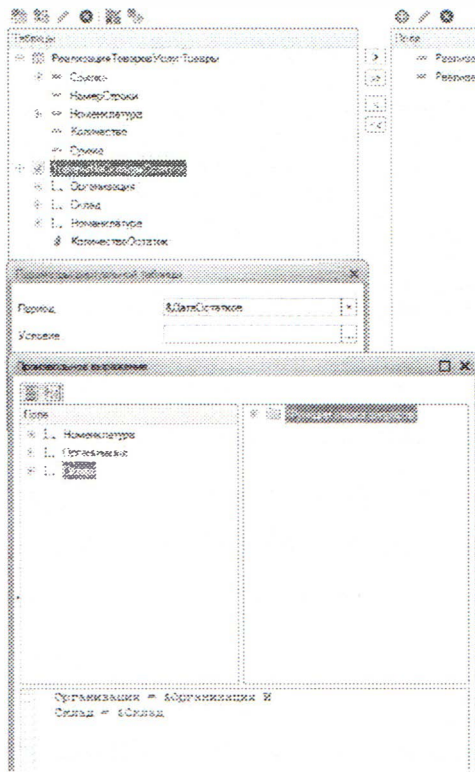


Рисунок 157

Из второго источника нам необходимо выбрать поле «КоличествоОстаток». Если мы прямо сейчас попробуем выполнить запрос, предварительно установив параметры, то увидим довольно странный результат (Рисунок 158-159).

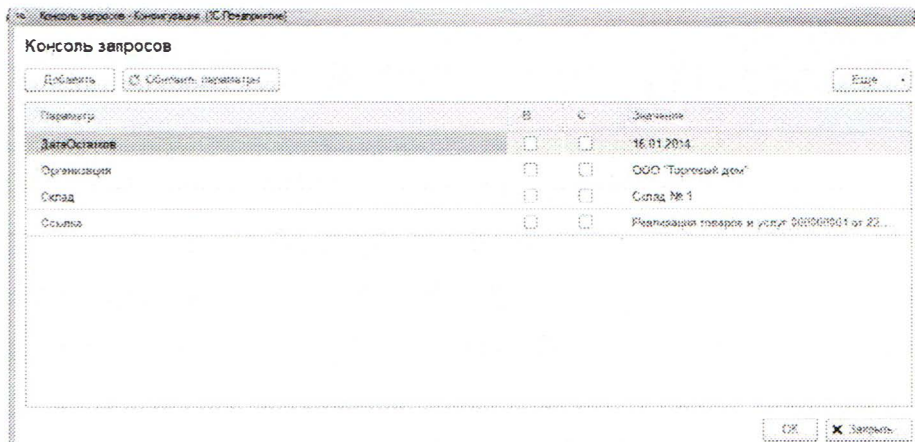


Рисунок 158

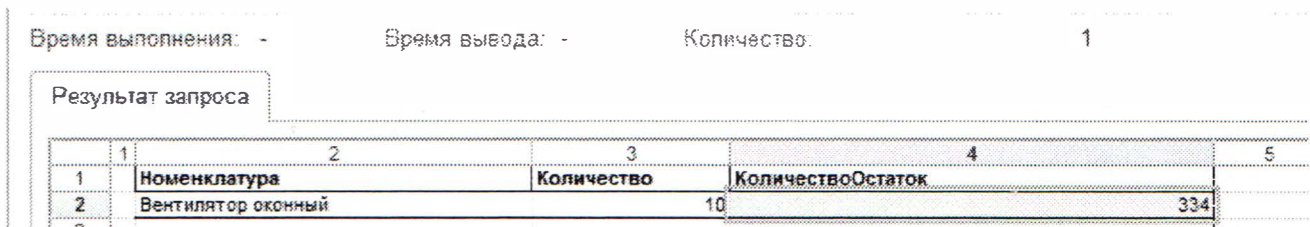


Рисунок 159

На самом деле мы получили не остаток по организации и складу, а остаток по всей номенклатуре.

Для того чтобы получить остаток именно по нашей позиции, нам необходимо связать обе таблицы по полю «Номенклатура». Сделаем мы это на закладке «Связи» конструктора запроса, которая появилась после добавления второй таблицы в запрос (Рисунок 160).



Рисунок 161

То есть, мы как бы сказали, что из таблицы нам нужны не все записи, а только с аналогичной номенклатурой. Вот теперь мы получили верный остаток на установленную дату по указанной организации и складу (Рисунок 167).

Время выполнения: - Время вывода: - Количество: 1

Результат запроса

1	2	3	4
1	Номенклатура	Количество	КоличествоОстаток
2	Вентилятор оконный	10	110
3			

Рисунок 161

2.13 Соединение таблиц теория.

Сейчас мы сделаем небольшое «лирическое» отступление и поговорим немного о теории. Чтобы понять, что собой представляет соединение таблиц и как оно происходит, нам необходимо рассмотреть небольшой пример. Ключевое слово здесь СОЕДИНЕНИЕ, то есть таблицы как бы «пристыковываются» друг к другу. А вот то, как между собой стыкуются отдельные строки обеих таблиц, мы и определяем на закладке «Связи» конструктора запроса. Представьте, что у нас есть 2 таблицы: табличная часть документа и таблица остатков (Таблица 1-2).

Таблица 1 – Документ

Номенклатура	Количество
Вилка	4
Ложка	6

Таблица 2 – Остатки

Номенклатура	Остаток
Табурет	12
Вилка	10

Нам требуется соединить эти таблицы между собой. Соединять мы их будем по номенклатуре, что вполне логично, так как поля, содержащие номенклатуру (они могут называться по-разному, главное, чтобы они содержали одинаковый тип данных), есть в обеих таблицах. И тут возникает вопрос: как быть со строками, которые есть в первой таблице

(имеется ввиду номенклатура), и нет во второй и, наоборот, как быть со строками, которые есть во второй таблице, но нет в первой? Вот тут могут быть разные варианты:

1. В результирующую таблицу попадают все строки из первой (левой) таблицы, независимо от того, есть ли им соответствие во второй таблице. Такой вариант называют «Левое внешнее соединение» или просто – «левое соединение».

Все просто: когда выводятся все строки из левой таблицы, то соединение левое.

Вот как соединятся наши таблицы при левом соединении (Таблица 3).

Таблица 3 – Левое соединение

Номенклатура	Количество	Остаток
Вилка	4	10
Ложка	6	NULL

Как видите, в таблицу вывелись все строки из левой таблицы. Строка из правой таблицы с номенклатурой «Вилка» присоединилась к левой таблице по условию равенства поля, содержащего номенклатуру, поэтому для этой строки есть остаток. Для строки с номенклатурой «Ложка» остатка нет, так как в правой таблице нет ни одной строки, содержащей номенклатуру «Ложка», поэтому система сама заполняет значение этого поля особым значением «NULL». «NULL» обозначает пустоту, то есть это даже не пустая ссылка, не ноль, не пустая строка, а именно отсутствие какого-либо значения. Это тоже надо запомнить. Левое соединение в конструкторе настраивается следующим образом (Рисунок 162).

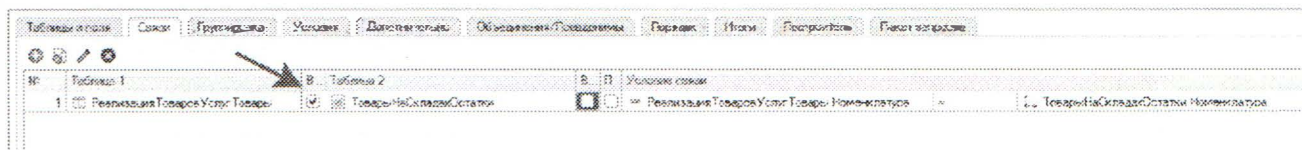


Рисунок 162

2. В результирующую таблицу попадают все строки из второй (правой) таблицы, независимо от того, есть ли им соответствие в первой (левой) таблице. Такой вариант называют «Правое внешнее соединение» или просто – «правое соединение».

Опять же: когда выводятся все строки из правой таблицы, то соединение правое.

Запомним и это. Вот так соединятся наши таблицы при правом соединении (Таблица 4).

Таблица 4 – Правое соединение

Номенклатура	Количество	Остаток
Табурет	NULL	12
Вилка	4	10

Здесь все аналогично: вывелись все строки из правой таблицы. Из левой таблицы присоединилась строка, содержащая номенклатуру «Вилка», поэтому заполнено поле «Количество». Для строки, содержащей номенклатуру «Табурет», нет соответствия в левой таблице, поэтому в поле количество здесь «NULL». В конструкторе это настраивается следующим образом (Рисунок 163).

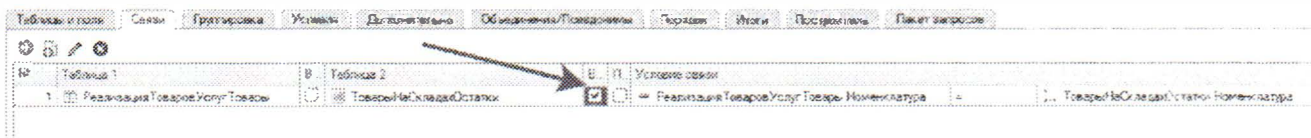


Рисунок 163

3. В результирующую таблицу попадают только те строки, для которых есть соответствие и в левой, и в правой таблицах. Другими словами, выводятся строки, которые удовлетворяют условию соединения. То есть, в нашем случае, в результат попадут только те строки, которые содержат одинаковую номенклатуру в первой (левой) и второй (правой) таблице. Такое соединение называется внутренним. Вот как соединятся наши таблицы при внутреннем соединении (Таблица 5).

Таблица 5 – Внутреннее соединение

Номенклатура	Количество	Остаток
Вилка	4	10

Тут все просто: строки, содержащие номенклатуру «Вилка» есть в первой и второй таблице. Они соединились в одну. Из первой таблицы выведены поля «Номенклатура» и «Количество», из второй – «Остаток». Настройка в конструкторе выглядит так (Рисунок 164).



Рисунок 164

4. В результирующую таблицу выводятся все строки из обеих таблиц. Такое соединение называется полным. То есть строки, отвечающие условию связи, соединятся

между собой, остальные строки дополняют результирующую таблицу. Вот как это будет выглядеть (Таблица 6).

Таблица 6 – Полное соединение

Номенклатура	Количество	Остаток
Вилка	4	10
Ложка	6	NULL
Табурет	NULL	12

Строки, содержащие номенклатуру «Вилка» соединились, строка с номенклатурой «Ложка» из первой таблицы и строка с номенклатурой «Табурет» из второй, так же попали в результат. Как это настраивается в конструкторе (Рисунок 165).



Рисунок 165

Стоит отметить, что конструктор «правое соединение» всегда преобразует в «левое». Просто он меняет таблицы местами, то есть левая становится правой, а правая – левой, соответственно и соединение становится левым. Таким образом, правое соединение легко превратить в левое, поменяв таблицы местами и поставив флажок «Все» у нужной таблицы. Можете проверить в конструкторе запросов, настроив правое соединение, посмотрев на текст запроса, или заново открыв конструктор.

2.14 Соединение таблиц. Пример.

Мы, наконец-то, добрались до финала нашего примера с соединением таблиц. В этой части посмотрим на наш пример на практике и выведем остатки для всей номенклатуры из документа, если таковые имеются. Итак, из документа нам нужны все строки, а из таблицы остатков – только удовлетворяющие условию соединения. Значит, мы используем левое соединение (Рисунок 166).



Рисунок 166

Для номенклатуры без остатков в поле «КоличествоОстаток» будет выведено значение «NULL», что, в нашем случае, обозначает нулевые остатки. Мы помним, что «NULL» – это особое значение, которое не равно нулю. Это плохо тем, что «NULL» очень
87 Верный старт в 1С www.work-1c.ru 88002344322 Бесплатный звонок

неудобно обрабатывать, если мы, например, захотим суммировать значения этого поля. Значит, нам надо как-то обработать этот случай и вместо «NULL» подставить «0». Такая возможность у нас есть благодаря функции «ЕСТЬNULL». Она позволяет проверить, равно ли поле значению «NULL» и, если равно, меняет на то значение, которое мы укажем вторым параметром. Как это работает, я продемонстрирую на конкретном примере.

В конструкторе запроса выделим поле «КоличествоОстаток» и нажмем кнопку редактирования (Рисунок 167).

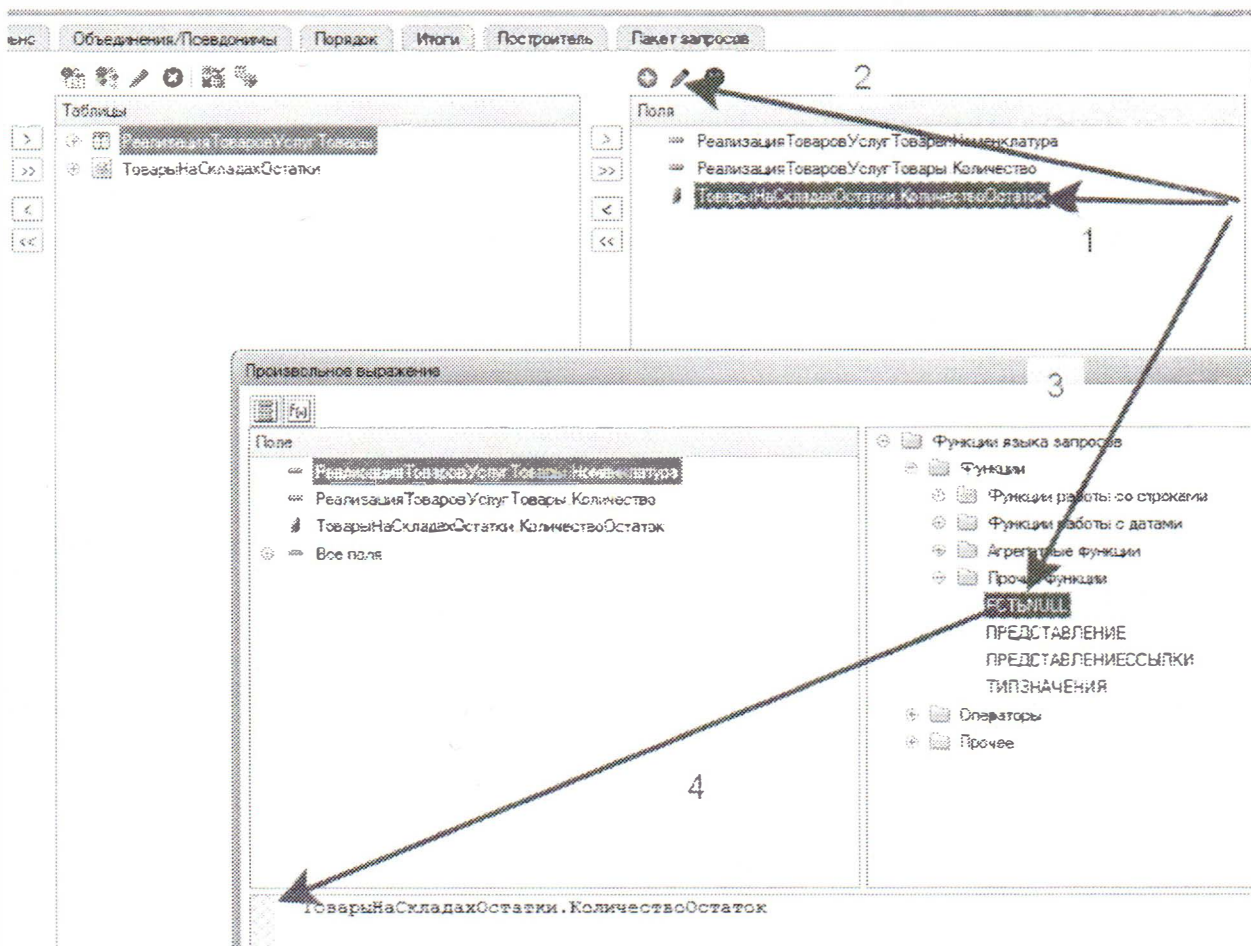


Рисунок 167

Выбираем в правом окне функцию «ЕСТЬNULL» и перетаскиваем ее в окно выражения, как показано стрелочкой №4, в начало строки. Итак, мы получили заготовку выражения, которое нам необходимо заполнить (Рисунок 168).

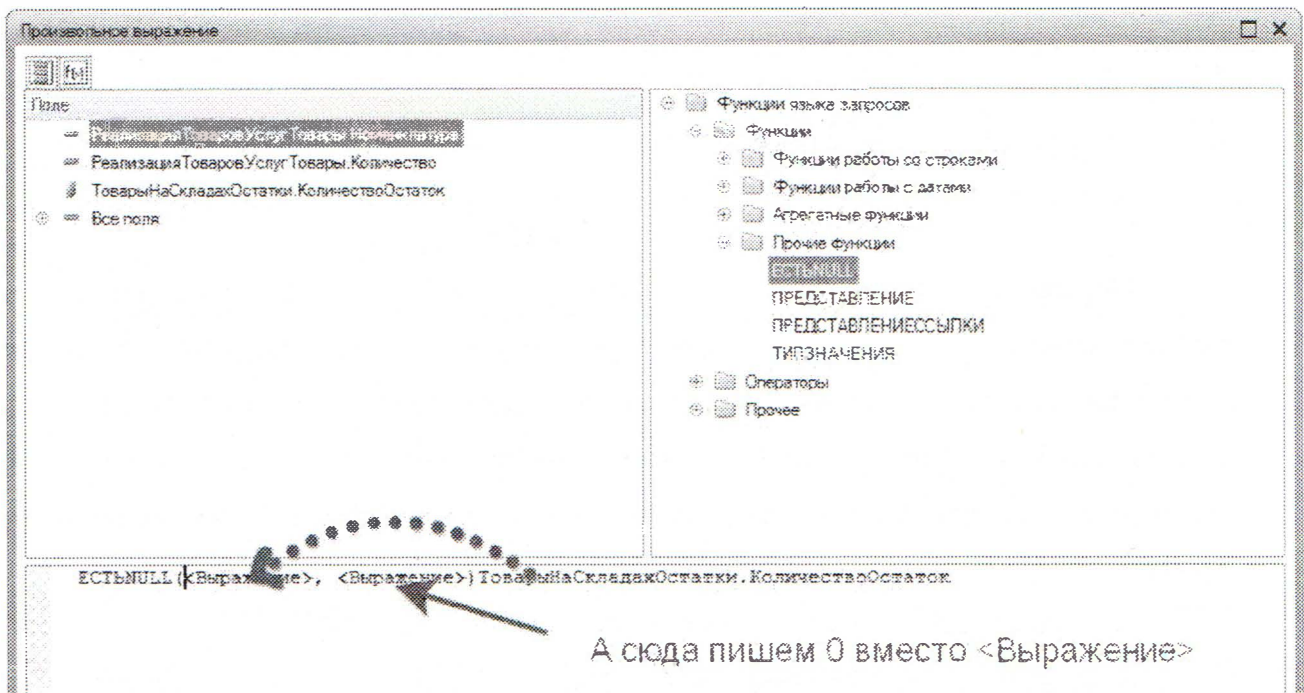


Рисунок 168

В итоге у нас должно получиться следующее (если наши планы наступления войск, вдруг, кого-то запутали) (Рисунок 169).

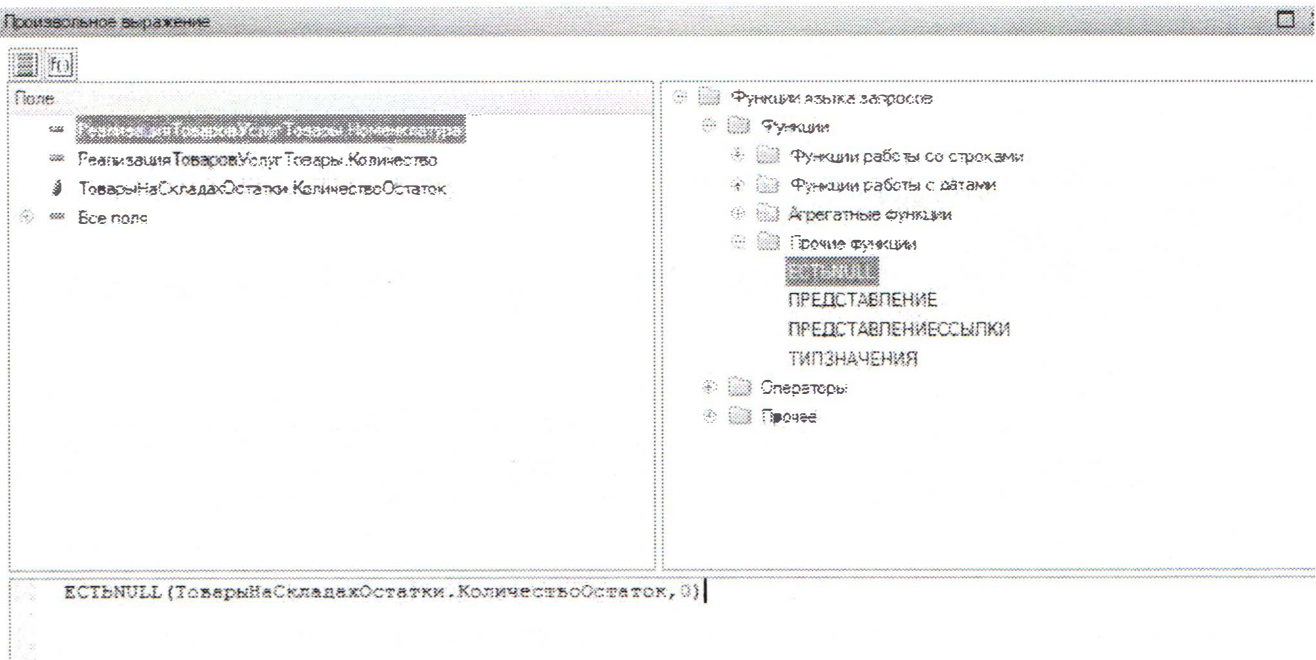


Рисунок 169

Мы как бы говорим: если у нас поле «КоличествоОстаток» равно «NULL», тогда нужно подставить ноль, иначе, пусть остается поле «КоличествоОстаток». Кроме того, после такого редактирования, система автоматически присвоит этому полю псевдоним «Поле1». Нас это не устраивает, и мы присвоим ему псевдоним «Остаток» (Рисунок 170).



Рисунок 170

Обрабатывать поле «Количество» на предмет значения «NULL» нет смысла, так как там его точно не будет, ведь мы берем все строки из таблицы документа. Хочу обратить ваше внимание на тот факт, что при внутреннем соединении проверять на NULL вообще нет смысла, так как в результате войдут только строки, отвечающие условию, то есть только совпадающие строки. А вот в случае полного соединения, любое из полей может принимать значение «NULL».

Не забываем про параметры нашего запроса (Рисунок 171).

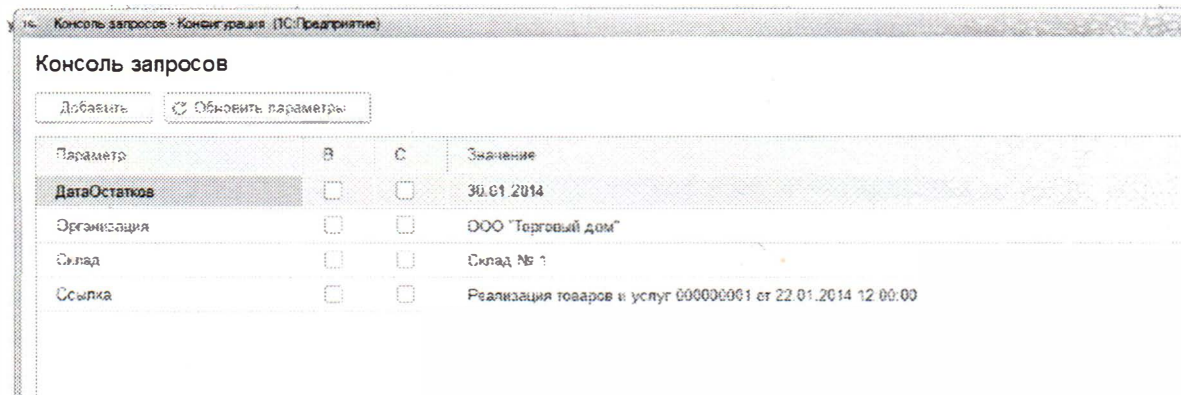


Рисунок 171

Выполним запрос и посмотрим на результат (Рисунок 172).

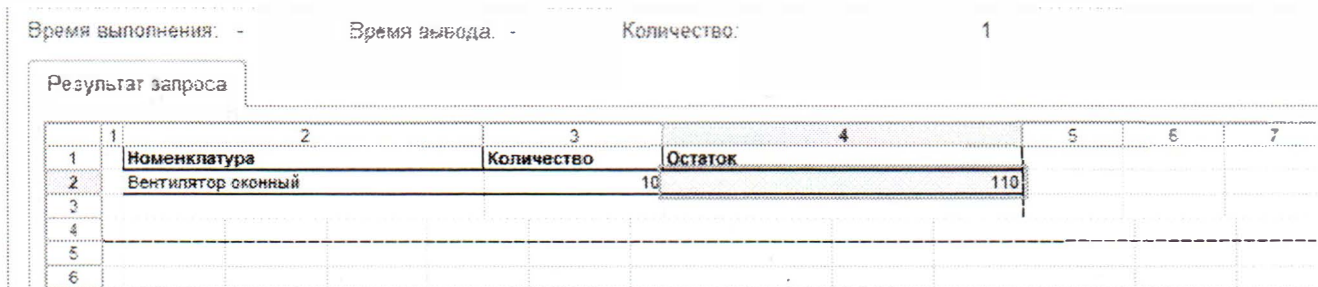


Рисунок 172

Теперь мы получили реальный остаток по товару из документа на заданную дату по организации и складу документа.

Глава 3.

3.1 Соединение таблиц. Детальный теоретический пример.

Как понятно из названия, в этой части мы продолжим обсуждение темы соединения таблиц. Только теперь мы рассмотрим этот вопрос более углубленно и детально. В

домашнем задании перед Вами стояла задача соединить виртуальные таблицы оборотов регистров накопления «Заказы покупателей» и «Продажи». Вот тут и возникает вопрос: как их связать? На конкретном примере рассмотрим, как будут вести себя соединяемые таблицы при тех или иных условиях. Для начала, скачайте информационную базу по ссылке <https://yadi.sk/d/CcZ64dvE-ILRjw>.

Итак, предположим, у нас есть 2 таблицы следующего вида (Рисунок 173).

Прод			Зак		
Склад	Номенклатура	Продано	Склад	Номенклатура	Заказано
Склад № 1	Вилка	4	Склад № 1	Табурет	12
Склад № 2	Ложка	6	Склад № 2	Вилка	10
			Склад № 1	Вилка	5

Рисунок 173

Слева располагается таблица продаж, справа – заказов. Давайте соединим таблицы по номенклатуре внутренним соединением (напомню, что в результате внутреннего соединения попадают только те строки обеих таблиц, которые удовлетворяют условию связи). В конструкторе запросов эта связь выглядела бы так (Рисунок 174).



Рисунок 174

Вот что мы получим, если соединим таблицы только по номенклатуре (Рисунок 175).

Прод			Зак		
Склад	Номенклатура	Продано	Склад	Номенклатура	Заказано
Склад № 1	Вилка	4	Склад № 1	Табурет	12
Склад № 2	Ложка	6	Склад № 2	Вилка	10
			Склад № 1	Вилка	5

✚ Внутреннее соединение по номенклатуре (только совпадающие)					
Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка	4	Склад № 2	Вилка	10
Склад № 1	Вилка	4	Склад № 1	Вилка	5

Рисунок 175

Строки, удовлетворяющие условию связи, то есть содержащие одинаковую номенклатуру, выделены одинаковым цветом. Именно эти строки попадают в результирующую таблицу. Как вы можете увидеть, результат состоит из ВСЕХ полей обеих таблиц, а мы уже выбираем из нее нужные. При таком условии связи строка из первой таблицы повторяется дважды, она продублирована. Если мы захотим получить итог по продажам, то он будет искажен. Кроме того, если мы будем брать поле «Склад» из таблицы заказов, то получится, что продажи есть и по первому складу, и по второму, хотя это не так. Получившаяся таблица содержит некорректные данные, так как наше условие связи является

недостаточным. Становится очевидным, что необходимо соединить таблицы не только по номенклатуре, но и по складу. Сделаем это.

В конструкторе это будет выглядеть так (Рисунок 176).



Рисунок 176

Вынужден сказать, что в случае с нашей базой, необходимо соединить и по организации, так как продажи и заказы могут быть по разным организациям. В примере с нашими таблицами организации нет, так что это не критично (Рисунок 177).

Прод			Зак		
Склад	Номенклатура	Продано	Склад	Номенклатура	Заказано
Склад № 1	Вилка	4	Склад № 1	Табурет	12
Склад № 2	Ложка	5	Склад № 2	Вилка	10
			Склад № 1	Вилка	5

Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка	4	Склад № 2	Вилка	10
Склад № 1	Вилка	4	Склад № 1	Вилка	5

Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка	4	Склад № 1	Вилка	5

Рисунок 177

Совпадающие строки выделены одинаковым цветом. Они и попали в результирующую таблицу. Теперь данные, содержащиеся в ней, правильные, искажений нет. строки не дублируются. Условие связи оказалось достаточным. Для упрощения примера мы рассмотрели внутренне соединение таблиц. Такое соединение не подойдет для решения поставленной задачи, так как часть данных о продажах и заказах теряется. Нам нужно вывести все сведения и о продажах, и о заказах. Как же быть? Нам потребуется полное соединение, тогда данные из обеих таблиц полностью попадут в результат.

В конструкторе полное соединение по номенклатуре настраивается так (Рисунок 178).

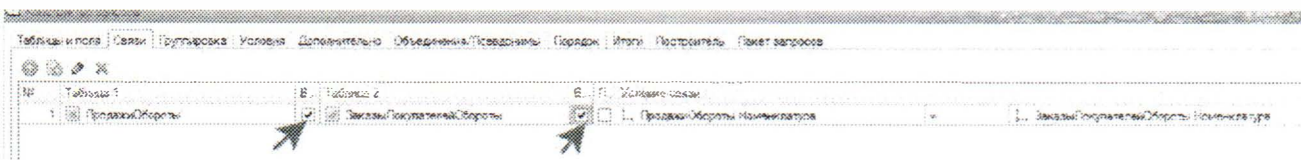


Рисунок 178

Нужно выставить флаг «Все» у обеих таблиц.

Так будет выглядеть результат при полном соединении по номенклатуре (Рисунок 179).

Полное соединение по номенклатуре

Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка	4	Склад № 2	Вилка	10
Склад № 1	Вилка		Склад № 1	Вилка	5
Склад № 2	Ложка	6	NULL	NULL	NULL
NULL	NULL	NULL	Склад № 1	Табурет	12

Рисунок 179

Мы видим те же строки, что и при внутреннем соединении. Кроме того добавились несовпадающие по номенклатуре строки каждой из таблиц. Для таблицы продаж это строка №2, для таблицы заказов – строка №1. У этих строк заполнены только те поля, которые взяты из исходной таблицы, остальные имеют значение NULL.

Идем дальше. Рассмотрим полное соединение по номенклатуре и складу в конструкторе запроса (Рисунок 180).

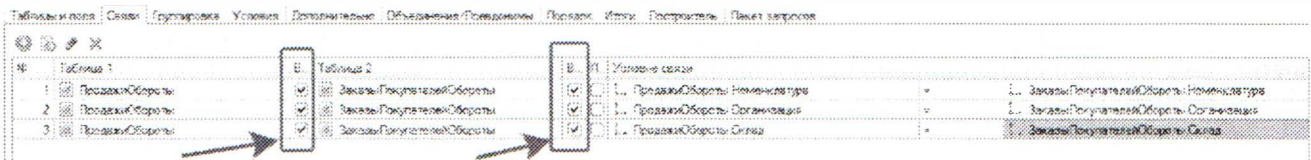


Рисунок 180

Результирующая таблица выглядит следующим образом (Рисунок 181).

Полное соединение по номенклатуре и складу

Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка		Склад № 1	Вилка	5
Склад № 2	Ложка	6	NULL	NULL	NULL
NULL	NULL	NULL	Склад № 2	Вилка	10
NULL	NULL	NULL	Склад № 1	Табурет	12

Рисунок 181

Как и при внутреннем соединении, мы имеем одно совпадение. Далее таблица дополнена несовпадающими строками из обеих таблиц: продажа ложки со склада №2, заказы вилки со склада №2 и табурета со склада №1. Таким образом, таблица для решения задачи готова, но нам мешают значения «NULL», которые усложняют обработку информации, не позволяют получить итоги и т.д. Мы с легкостью выйдем из этого затруднения, используя функцию «ЕстьNULL» языка запросов, как мы это делали в части 14 урока 2. Вот так выглядят выражения, которые мы будем использовать (Рисунок 182).

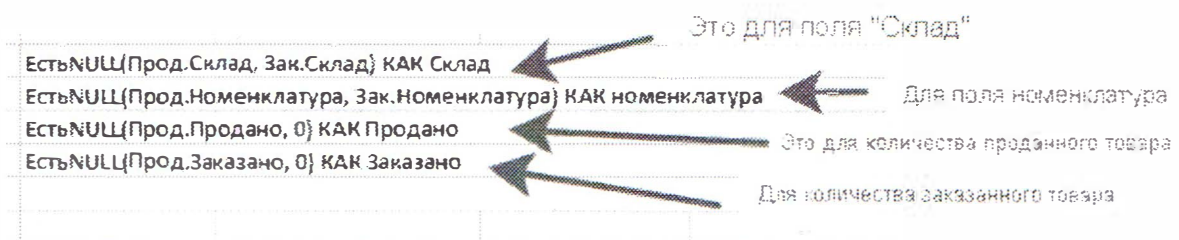


Рисунок 182

Немного поясню. Для склада: берем склад из таблицы продаж, если же он принимает значение «NULL», то берем склад из таблицы заказов. Для номенклатуры – аналогично. Для количества проданного товара: если оно принимает значение «NULL», то подставляем 0. Точно так же поступаем с количеством заказанной номенклатуры.

Еще один важный нюанс соединения таблиц. Касается он дублирования строк. Представьте, что мы возьмем не виртуальную таблицу заказов, а реальную. Данные в ней не сгруппированы и выглядят так (Рисунок 183)

Прод			Зак		
Склад	Номенклатура	Продано	Склад	Номенклатура	Заказано
Склад № 1	Вилка	4	Склад № 1	Табурет	12
Склад № 2	Ложка	6	Склад № 1	Вилка	10 Документ 1
			Склад № 1	Вилка	5 Документ 2

Рисунок 183

Так, даже соединив по номенклатуре и складу, мы получим дубли строк (строки, удовлетворяющие условию связи, выделены одинаковым цветом) (Рисунок 184).

Прод			Зак		
Склад	Номенклатура	Продано	Склад	Номенклатура	Заказано
Склад № 1	Вилка	4	Склад № 1	Табурет	12
Склад № 2	Ложка	6	Склад № 1	Вилка	10 Документ 1
			Склад № 1	Вилка	5 Документ 2

Внутреннее соединение по номенклатуре и складу (только совпадающие)

Прод.Склад	Прод.Номенклатура	Прод.Продано	Зак.Склад	Зак.Номенклатура	Зак.Заказано
Склад № 1	Вилка	4	Склад № 1	Вилка	10
Склад № 1	Вилка	4	Склад № 1	Вилка	5

Рисунок 184

Для того чтобы избежать такой ситуации, данные необходимо предварительно сгруппировать. В нашем примере необходимо сгруппировать таблицу заказов по складу и номенклатуре. Сгруппированная таблица тогда бы выглядела так (Рисунок 185).

Склад	Номенклатура	Заказано
Склад № 1	Табурет	12
Склад № 1	Вилка	15

Рисунок 185

Только так при соединении таблиц мы получим верные данные.

3.2 Объединение таблиц.

При объединении таблиц одна таблица как-бы продолжает другую. Если при соединении таблиц мы «пристыковываем» одну таблицу к другой слева или справа от нее

(увеличивается количество полей), то при объединении мы «пристыковываем» одну таблицу под другой. Количество полей и их псевдонимы совпадают (Рисунок 186).

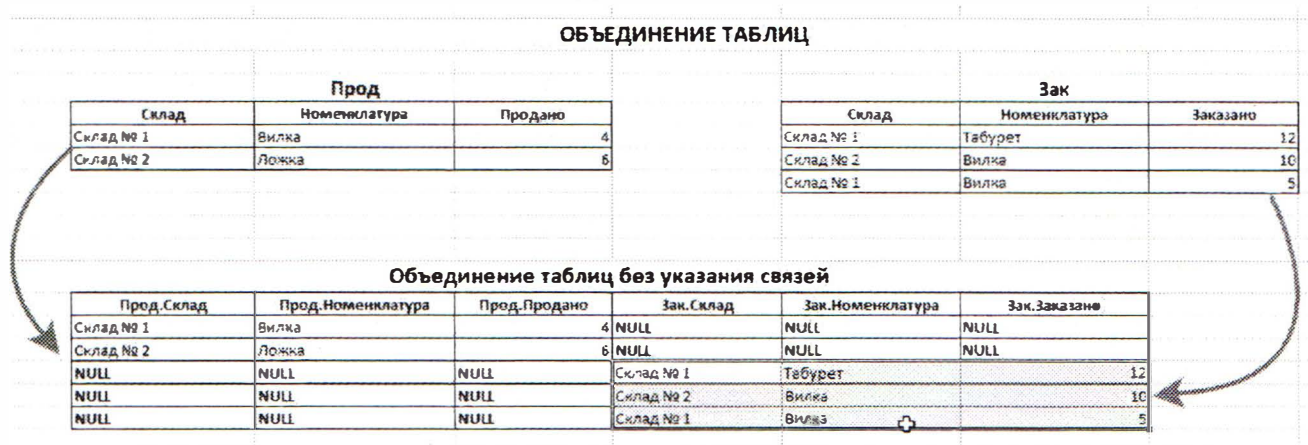


Рисунок 186

В таком виде с таблицей неудобно работать: она не наглядна и не информативна. Вот если бы мы могли указать, что склад из первой таблицы – это то же самое, что и склад из второй, так же, как и номенклатура из первой таблицы – то же самое, что номенклатура из второй, то тогда результат бы выглядел так (Рисунок 187).

Объединение таблиц с указанием связей по одинаковым реквизитам

Склад	Номенклатура	Прод.Продано	Зак.Заказано
Склад № 1	Вилка	4	NULL
Склад № 2	Ложка	6	NULL
Склад № 1	Табурет	NULL	12
Склад № 2	Вилка	NULL	10
Склад № 1	Вилка	NULL	5

Рисунок 187

Теперь наша таблица очень наглядна и понятна. Если же мы обработаем значение «NULL» так, как делали это в предыдущих уроках, то получим такую таблицу (Рисунок 188).

Объединение таблиц с указанием связей по одинаковым реквизитам значений для несовпадающих

Склад	Номенклатура	Прод.Продано	Зак.Заказано
Склад № 1	Вилка	4	0
Склад № 2	Ложка	6	0
Склад № 1	Табурет	0	12
Склад № 2	Вилка	0	10
Склад № 1	Вилка	0	5

Рисунок 188

То есть, для несовпадающих полей, мы указали, что значение будет 0.

Давайте посмотрим, как это делается в конструкторе запроса. Для запроса выберем те же таблицы, что и ранее в 1-й части (Рисунок 189).

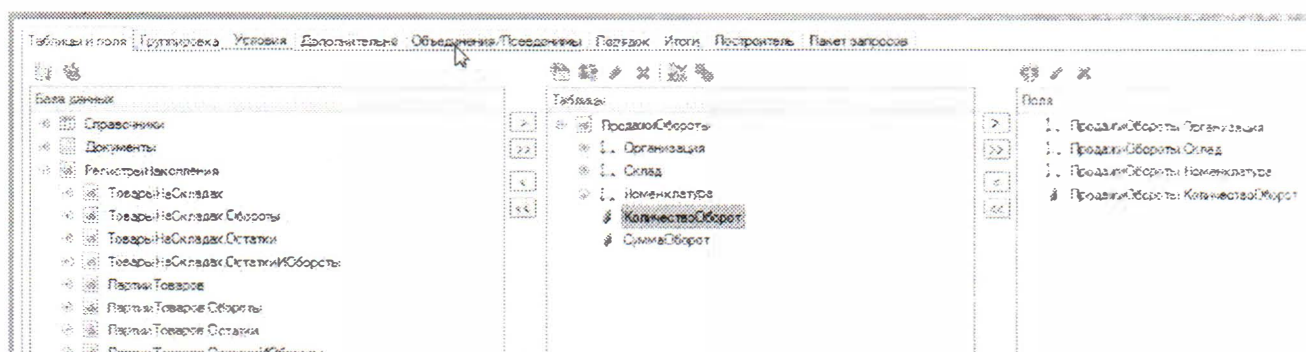


Рисунок 189

Сначала выбираем поля из одной таблицы, затем переходим на закладку «Объединения/Псевдонимы». Ранее на этой вкладке мы только задавали псевдонимы, теперь мы создадим здесь объединение таблиц. Для этого в левой части окна мы выделим строку «Запрос1» (это и есть наш текущий запрос в объединении). Теперь активны кнопки «Добавить» и «Скопировать» (Рисунок 190).

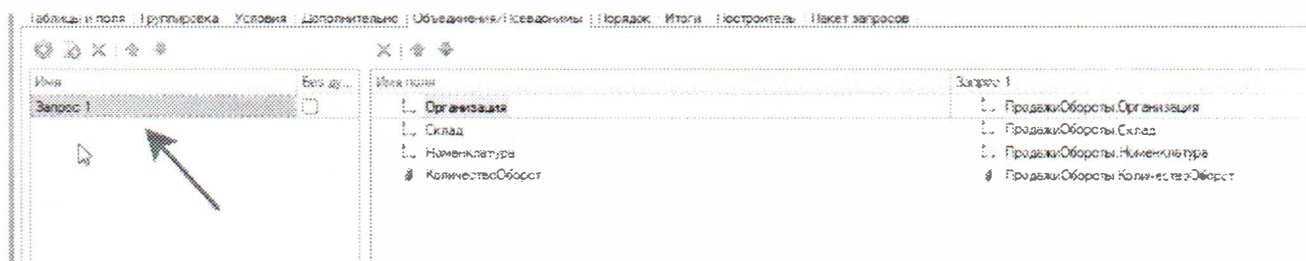


Рисунок 190

Нажав кнопку «Добавить» или выполнив аналогичное действие из контекстного меню, добавляем новый запрос и оказываемся на закладке «Таблицы и поля». На самом деле мы оказались в новом запросе, который является частью нашего объединения. Здесь мы выбираем вторую таблицу и поля для объединения (Рисунок 191-193).

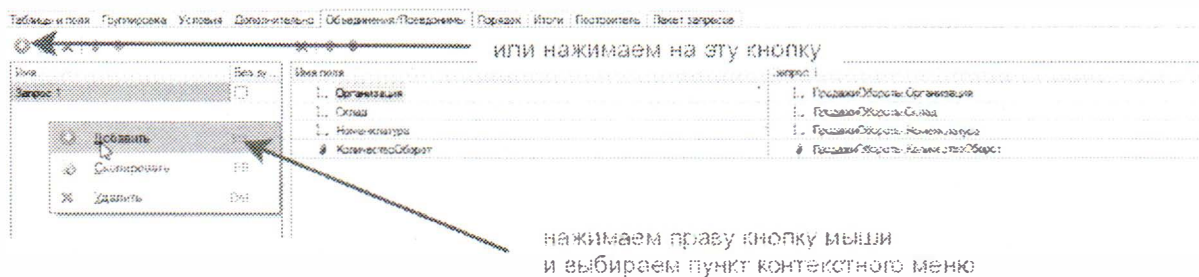


Рисунок 191

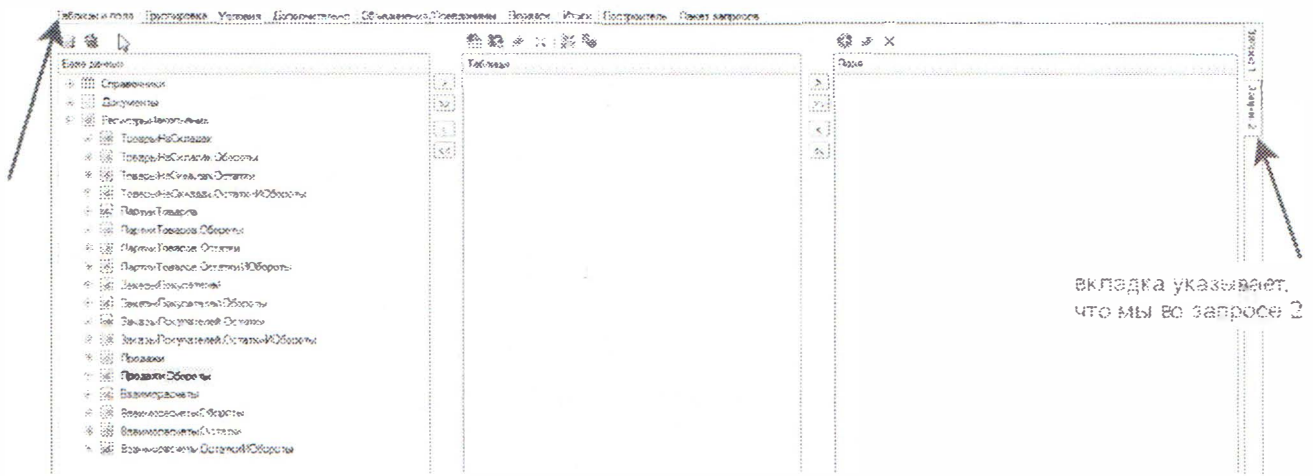


Рисунок 192

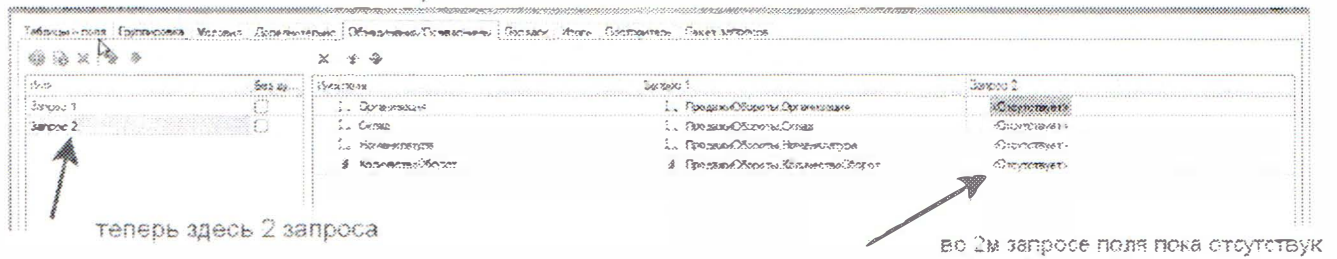


Рисунок 193

Выбираем таблицу и поля второго запроса (Рисунок 194).

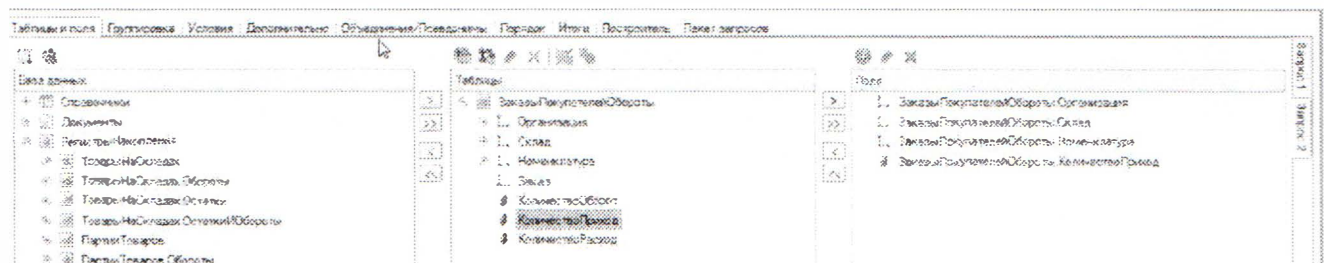


Рисунок 194

Переходим на закладку «Объединения/Псевдонимы» (Рисунок 195).

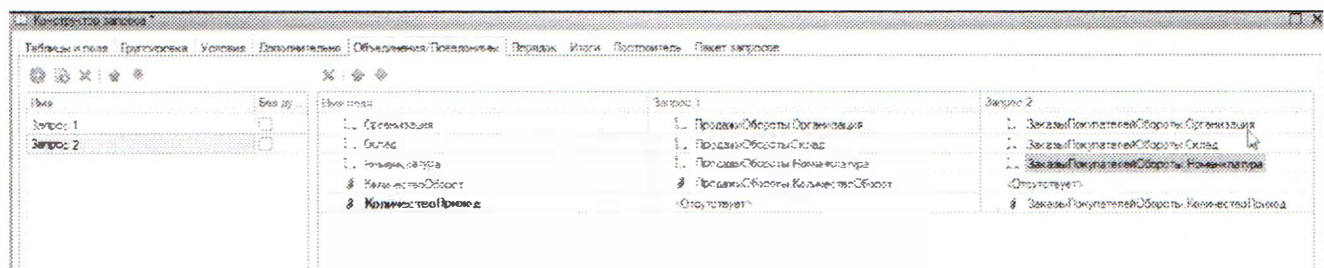


Рисунок 195

Видим, что поля из запроса 1 сопоставились с полями запроса 2. Это произошло благодаря тому, что их наименования совпадают, но они могли и не сопоставиться, тогда сопоставление можно было бы выполнить вручную, выбрав нужное поле в соответствующей колонке. Если бы связи не установились, то поля на закладке «Объединения/Псевдонимы» выглядели бы так (Рисунок 196).

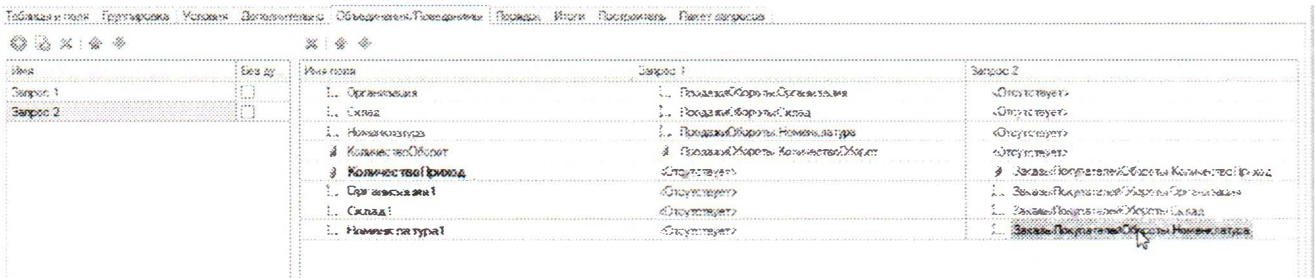


Рисунок 196

Выбирая поля в соответствующей колонке, соответствие можно назначить вручную (Рисунок 197).

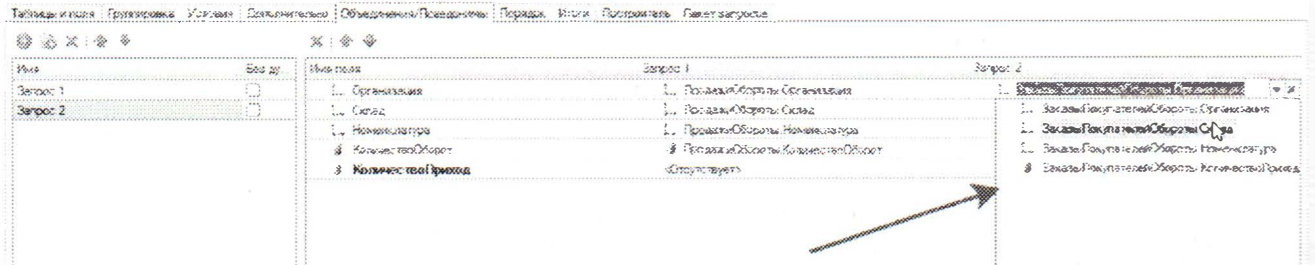


Рисунок 197

Переименуем поля «КоличествоОборот» в «Продано», а «КоличествоПриход» в «Заказано» (Рисунок 198).

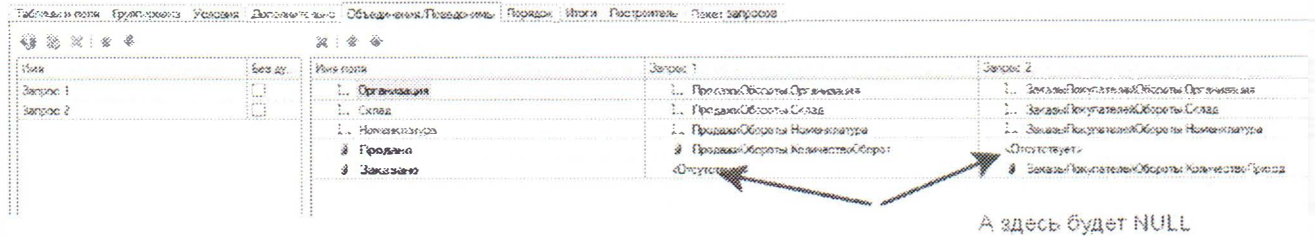


Рисунок 198

Выполним запрос, посмотрим на результат (Рисунок 199).

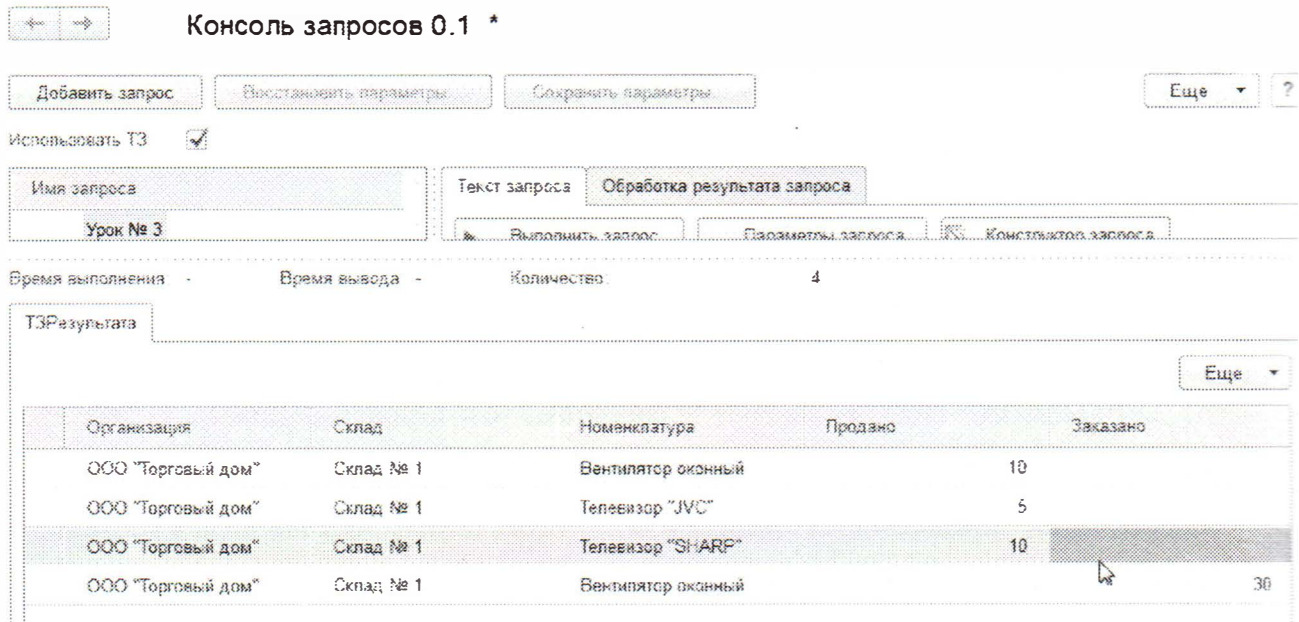


Рисунок 199

К сожалению, значения «NULL» не отображаются. Вместо них нам необходимо подставить нули. Для этого мы просто добавляем поле в каждом запросе нашего объединения и значение этого поля – «0» (Рисунок 200-201).

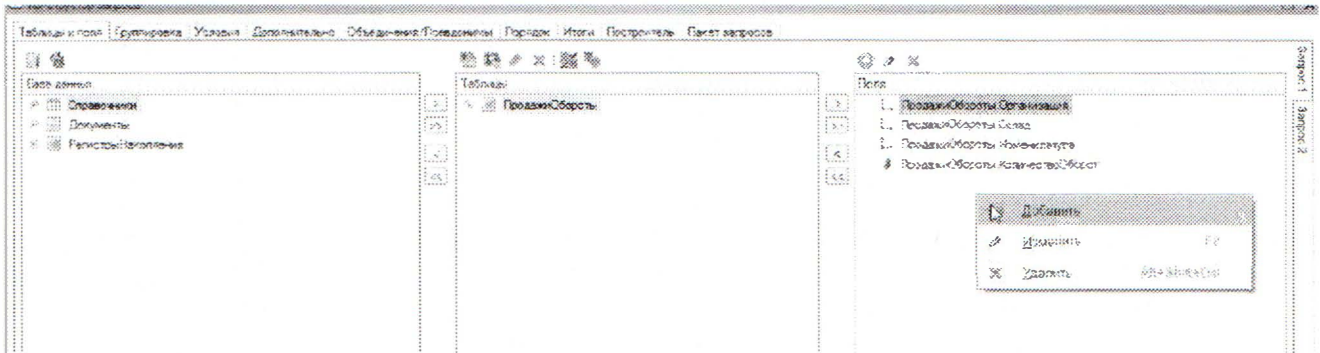


Рисунок 200

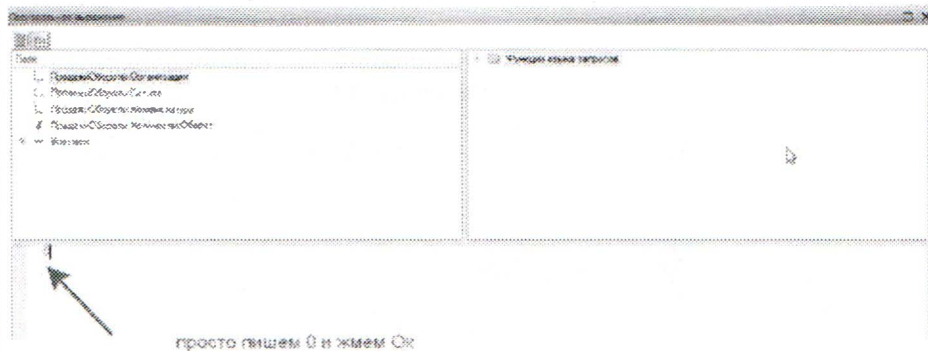


Рисунок 201

Это нужно сделать в обоих запросах нашего объединения. Затем на вкладке «Объединения/Псевдонимы» выбираем эти поля там, где они не сопоставились (Рисунок 202-203).

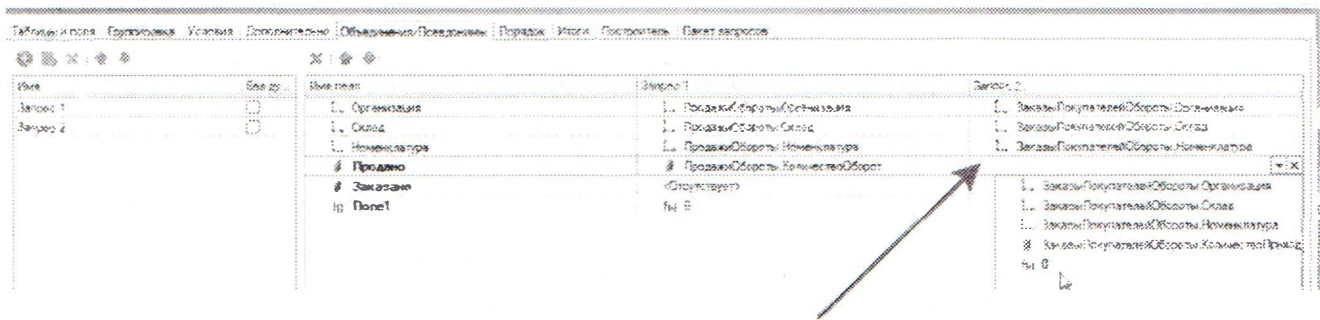


Рисунок 202

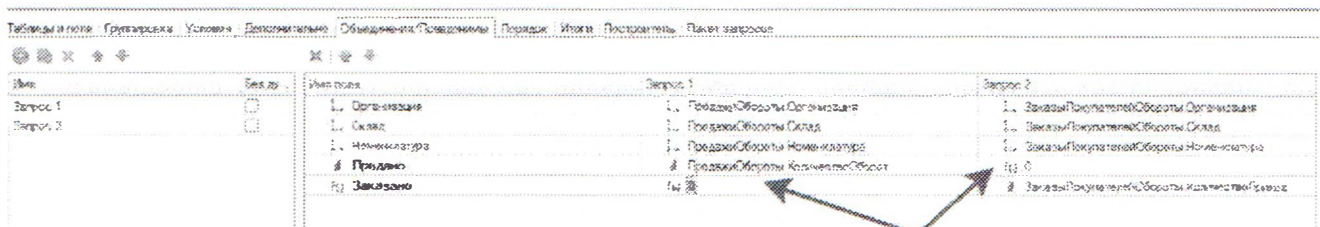


Рисунок 203

Выполним запрос. Отметим, что «0», так же, как и «NULL», не отображается в результате, но если мы дважды щелкнем по пустой ячейке, то увидим окно, в котором отображается «0» (Рисунок 204).

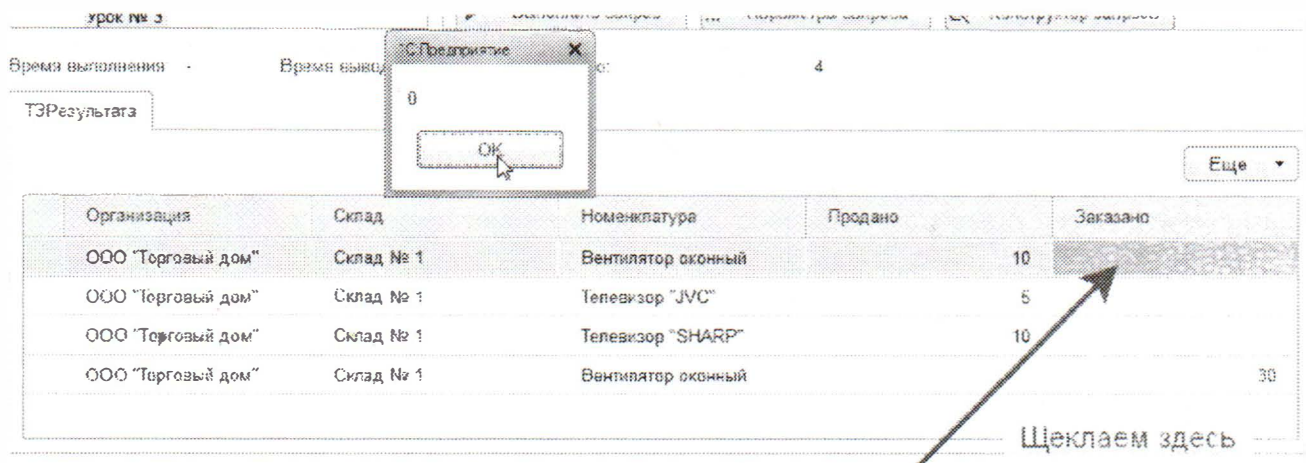


Рисунок 204

До того, как мы добавили поля с выражением «0» и сопоставили их, было так (Рисунок 205).

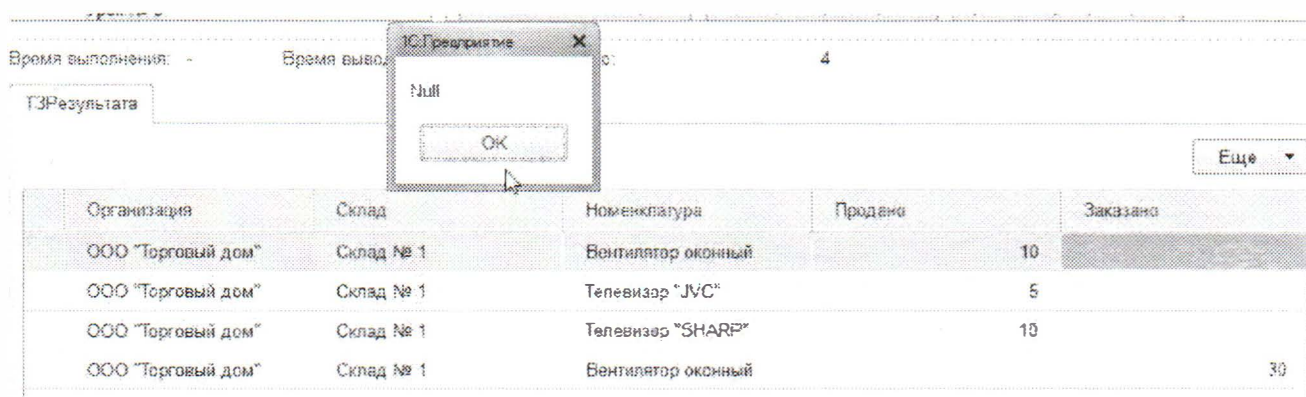


Рисунок 205

Объединение таблиц, в отличие от соединения, позволяет избежать дублей строк, как это было в предыдущем уроке.

3.3 Вложенные запросы.

Если мы внимательнее посмотрим на наш запрос, то увидим, что у нас есть строки по одной и той же номенклатуре, одному и тому же складу, и одной и той же организации, но в одной строке заказано, в другой – продано (Рисунок 206).

Организация	Склад	Номенклатура	Продано	Заказано
ООО "Торговый дом"	Склад № 1	Вентилятор оконный	10	
ООО "Торговый дом"	Склад № 1	Телевизор "JVC"	5	
ООО "Торговый дом"	Склад № 1	Телевизор "SHARP"	10	
ООО "Торговый дом"	Склад № 1	Вентилятор оконный		30

Рисунок 206

Это происходит по причине того, что наша таблица не сгруппирована. Давайте попробуем это сделать. Нам придется в каждом запросе объединения выполнить группировку (Рисунок 207).

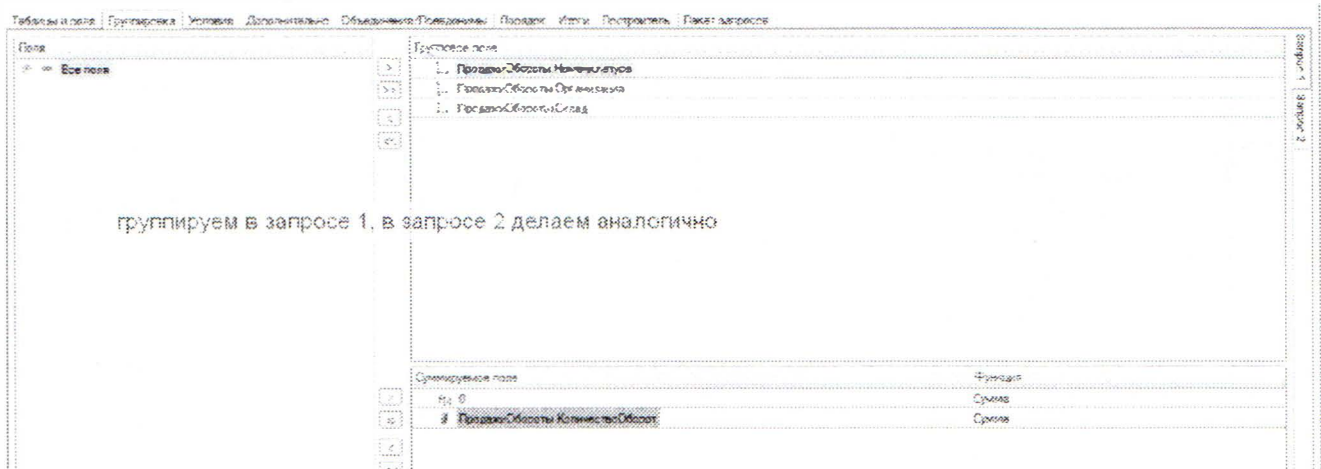


Рисунок 207

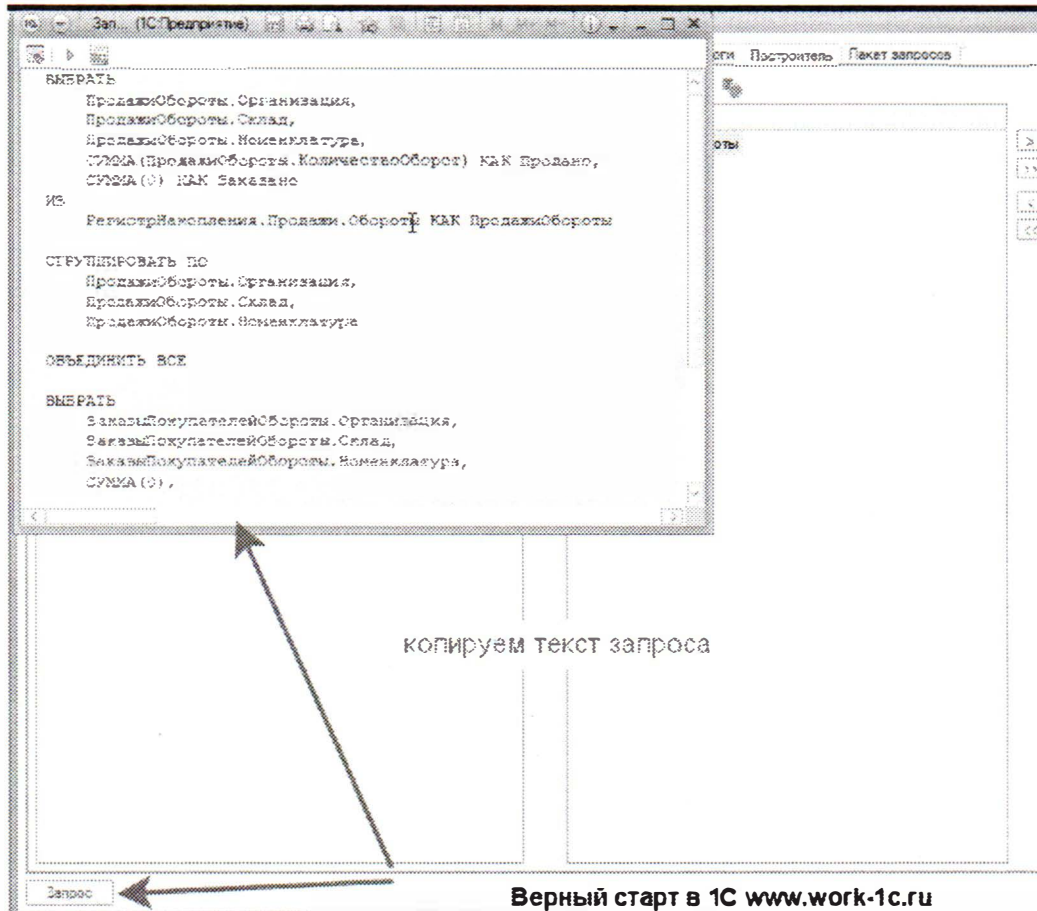
Однако, если выполнить такой запрос, то результат не изменится, ведь мы сгруппировали результаты каждого запроса нашего объединения по-отдельности, а нам необходимо сгруппировать все объединение. Как же быть? Здесь нам поможет вложенный запрос. Вложенный запрос – это запрос внутри запроса, то есть запрос верхнего уровня получает результат вложенного запроса и его можно дальше обрабатывать. Для того чтобы создать вложенный запрос достаточно на закладке «Таблицы и поля» нажать кнопку «Создать вложенный запрос» (Рисунок 208).



Рисунок 208

После этого откроется окно конструктора запроса, в котором можно создать вложенный запрос. Сам процесс создания вложенного запроса практически ничем не отличается от создания обычного запроса, за исключением того, что в нем нет итогов.

Если мы сейчас попытаемся создать вложенный запрос, то существующий запрос будет запросом верхнего уровня, а нам необходимо, чтобы он стал вложенным, то есть наше объединение таблиц должно стать вложенным запросом. Как это сделать? Хитрость заключается в том, чтобы скопировать текст существующего запроса и в нем создать вложенный запрос, а текст скопированного запроса станет текстом вложенного запроса. На практике это будет выглядеть так (Рисунок 209).



Нажимаем на кнопку "Создать вложенный запрос"

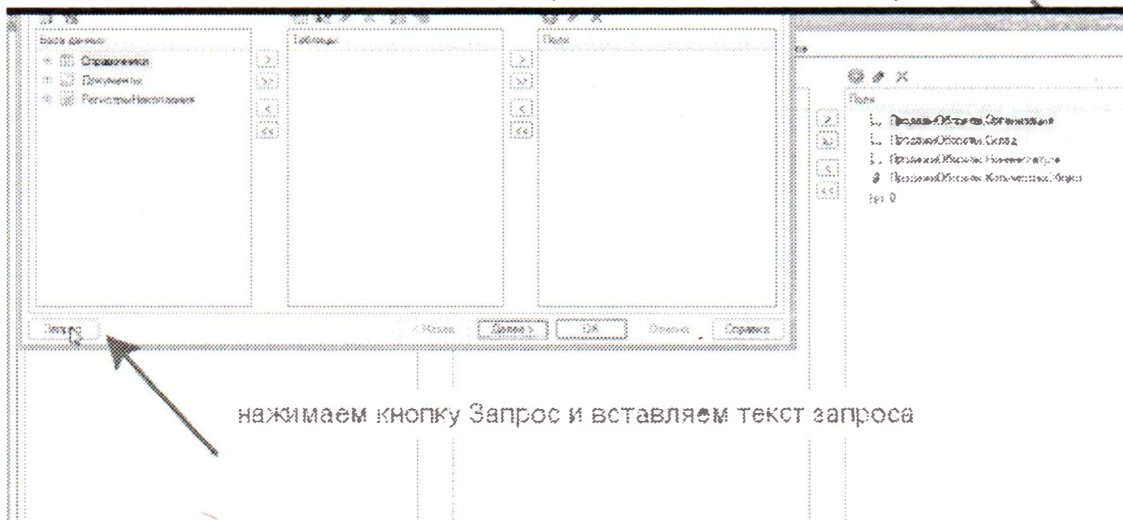


Рисунок 209

После этого сформировался вложенный запрос (Рисунок 210).

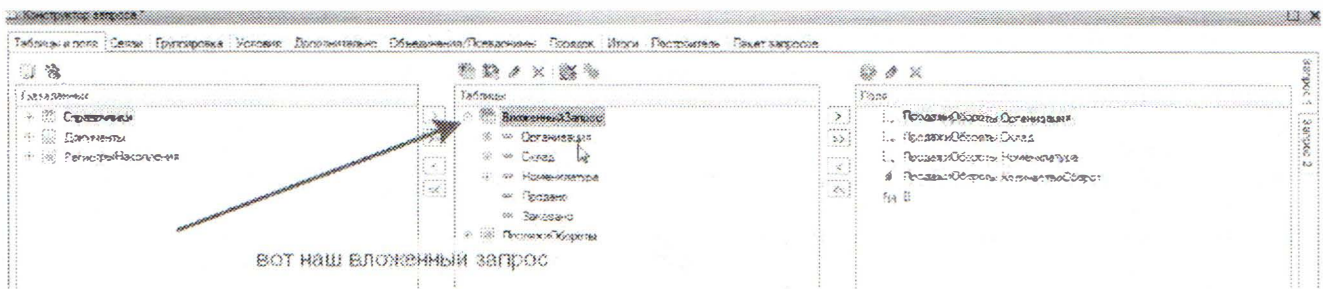


Рисунок 210

Он содержит те же поля, что и наш первоначальный запрос с объединением таблиц. По сути, исходный запрос нам уже не нужен и нам необходимо удалить запрос2 нашего объединения и таблицу «ПродажиОбороты» (Рисунок 211).

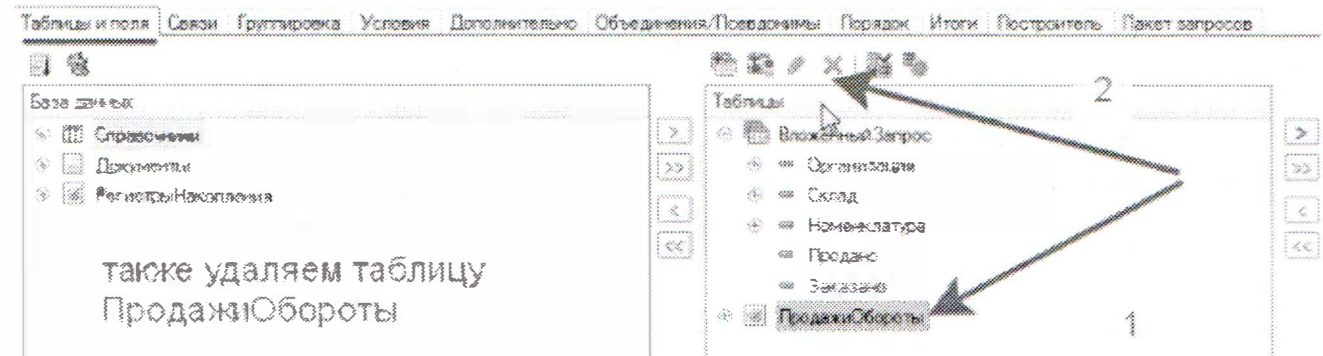
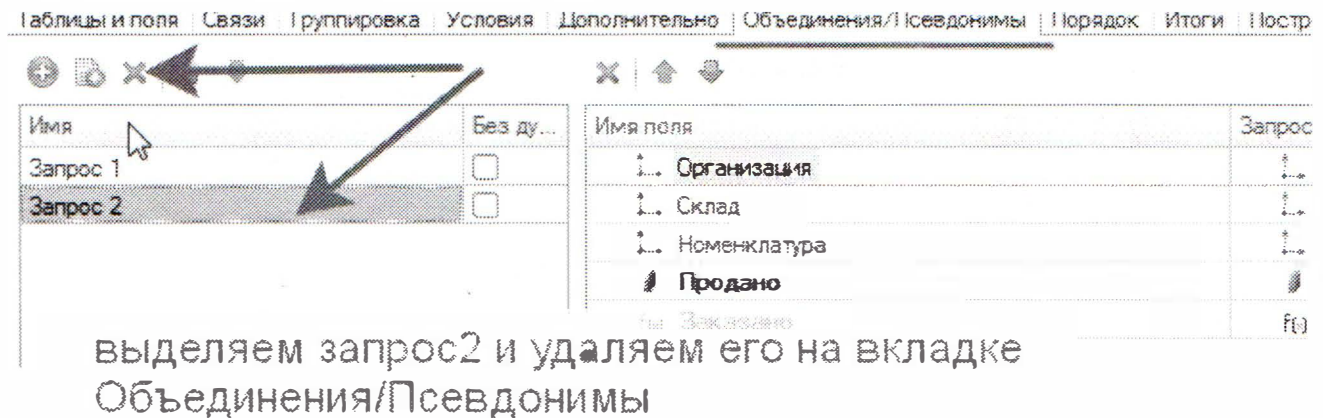


Рисунок 211

Теперь весь запрос, который мы ранее создали, находится во вложенном запросе. Можем это проверить, нажав на кнопку «Изменить» и выделив таблицу вложенного запроса на вкладке «Таблицы и поля» (Рисунок 212).

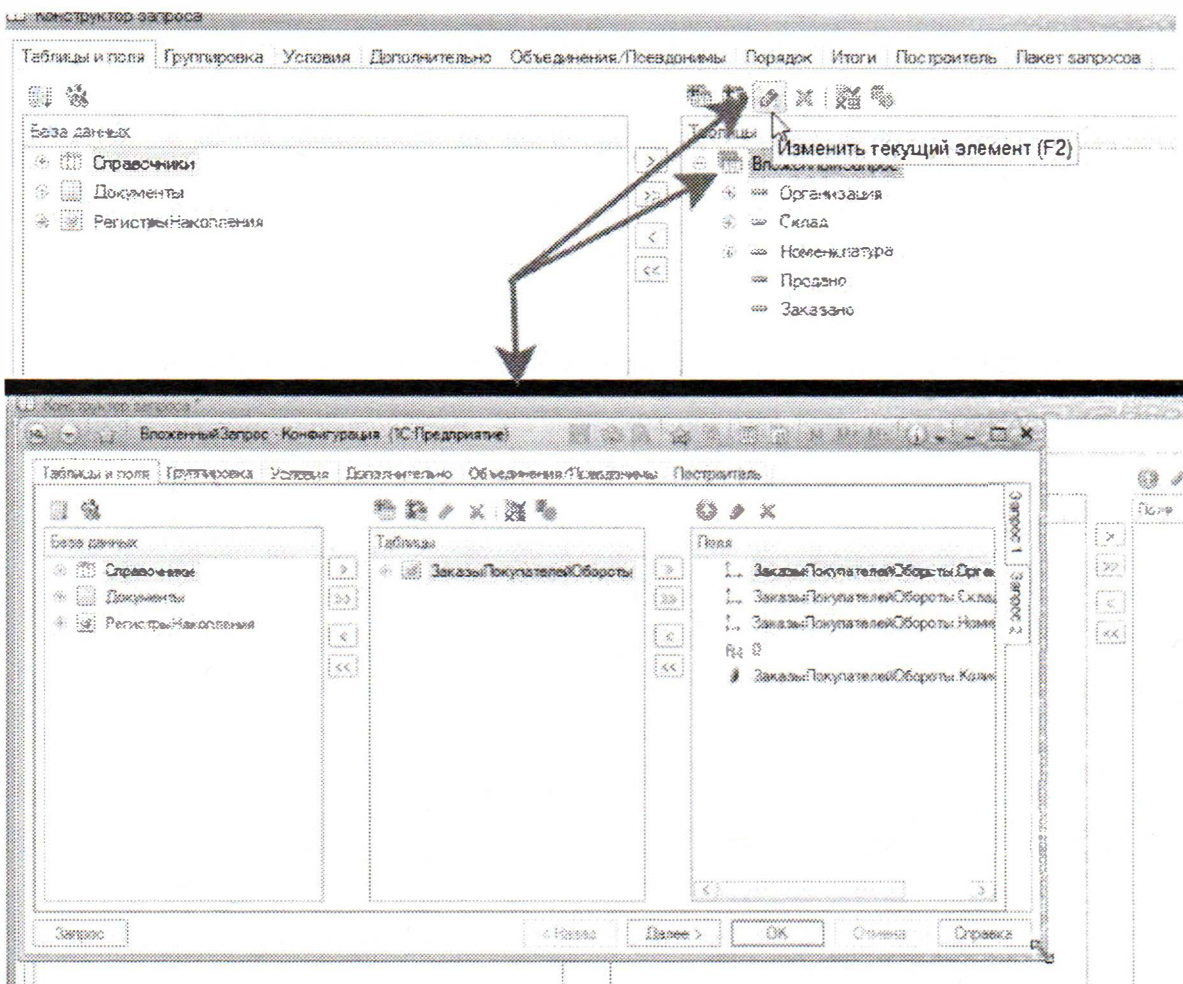


Рисунок 212

Также мы можем переименовать наш вложенный запрос (Рисунок 213).

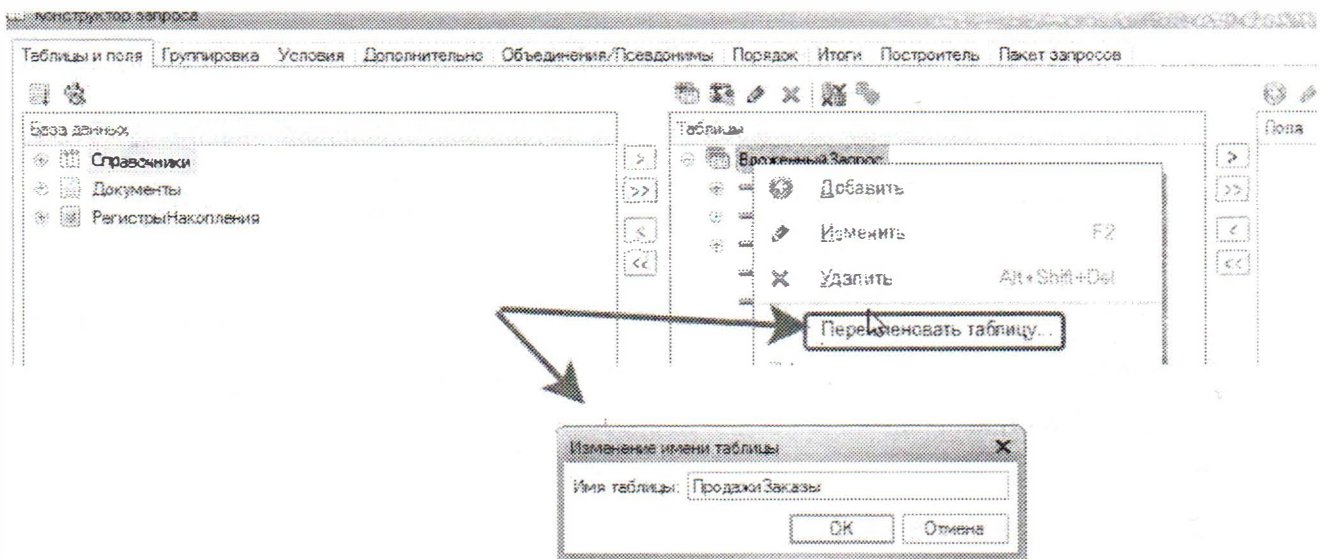


Рисунок 213

Теперь выбираем из вложенного запроса все поля и группируем таблицу (Рисунок 214).

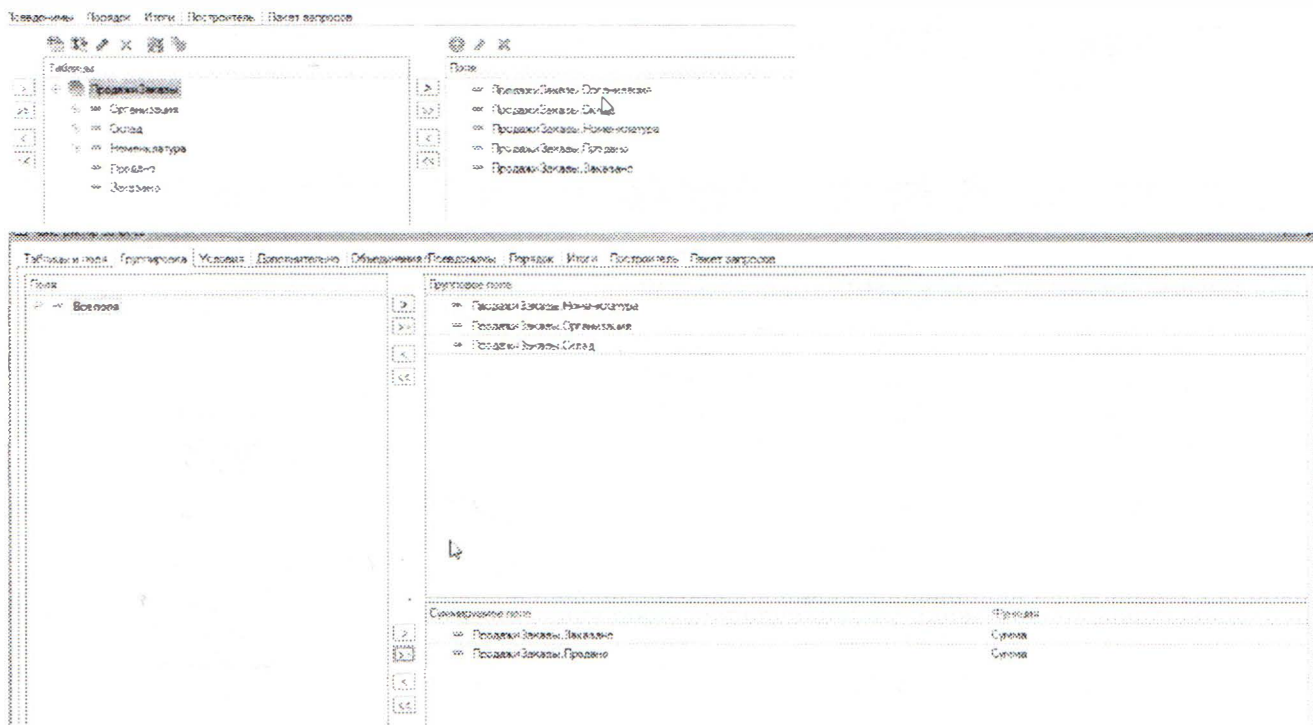


Рисунок 214

Теперь еще раз о том, что мы сделали: прежний запрос-объединение мы поместили во вложенный запрос. В результате, вложенный запрос получил несгруппированную таблицу-объединение и уже теперь единственное, что мы делаем с вложенным запросом, это просто группируем его. Вот как теперь выглядит результат (Рисунок 215).

ТЗРезультата

Организация	Склад	Номенклатура	Продано	Заказано
ООО "Торговый дом"	Склад № 1	Вентилятор оконный	10	30
ООО "Торговый дом"	Склад № 1	Телевизор "JVC"	5	
ООО "Торговый дом"	Склад № 1	Телевизор "SHARP"	10	

Рисунок 215

«Вентилятор оконный» теперь сгруппировался в одну строку, а цифры заказов и продаж расположились рядом. Результат очень похож на соединение таблиц, которое мы выполняли в предыдущем уроке, не так ли?

Необходимо отметить, что в терминах языка запросов, вложенный запрос – это таблица, точно такая же, как и любая другая реальная таблица. То есть вложенные запросы можно соединять или объединять с другими таблицами базы данных, накладывать условия и выполнять все действия, которые мы выполняем с другими таблицами. Из вложенного запроса мы не видим данных, присутствующих во внешнем запросе (запросе верхнего уровня), зато в запросе верхнего уровня мы видим все выходные поля вложенного запроса.

3.4 Функции языка. Строковые.

До сих пор в конструкторе запроса мы брали поля из таблиц базы данных. Однако, в языке запросов существует возможность добавлять свои поля с произвольными выражениями (Рисунок 216).

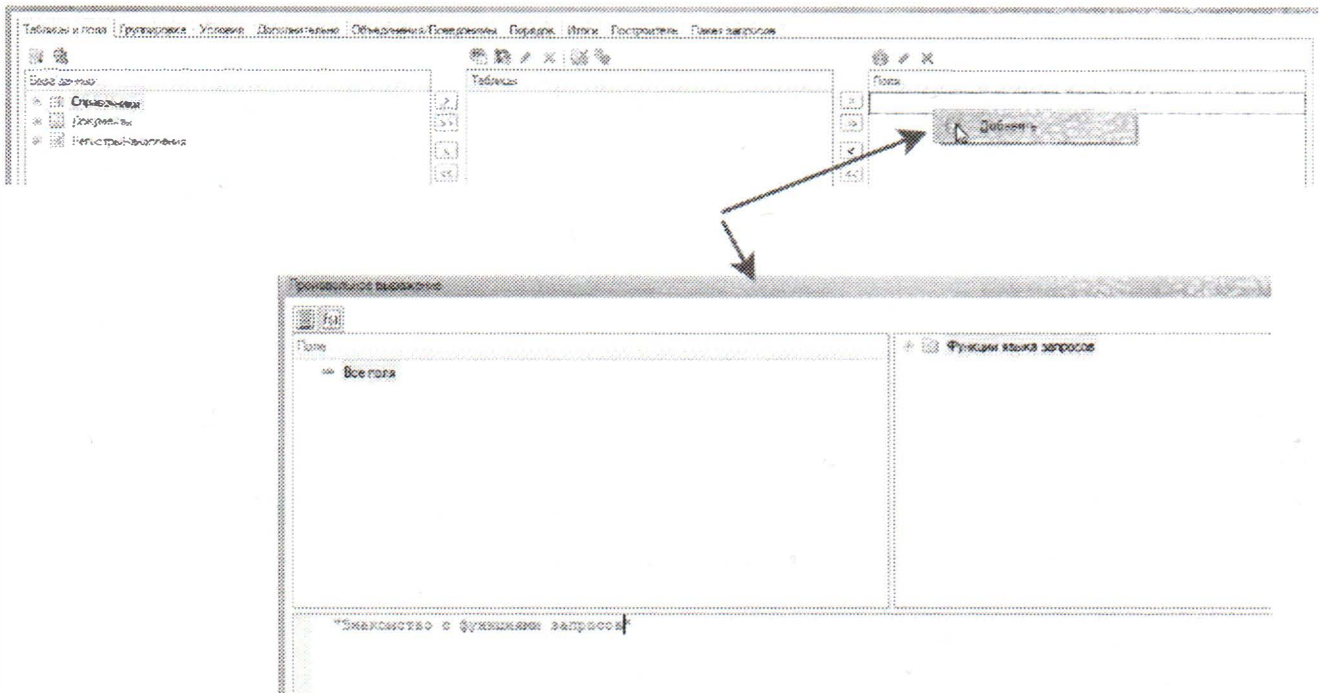


Рисунок 216

В данном случае мы просто создали поле со строкой. В языке запросов строка обрамляется кавычками.

Выполним запрос и посмотрим, что получилось (Рисунок 217).

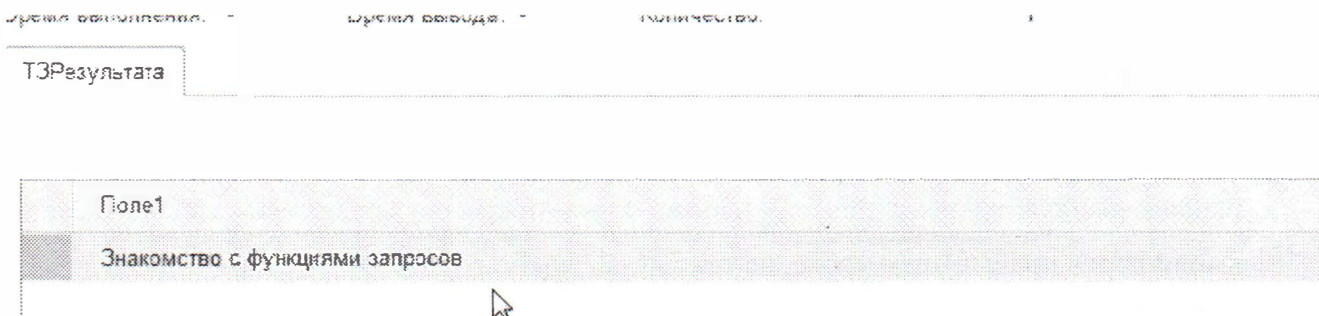


Рисунок 217

Мы получили единственное поле со строкой. Полю можно присвоить более информативный псевдоним (Рисунок 218).

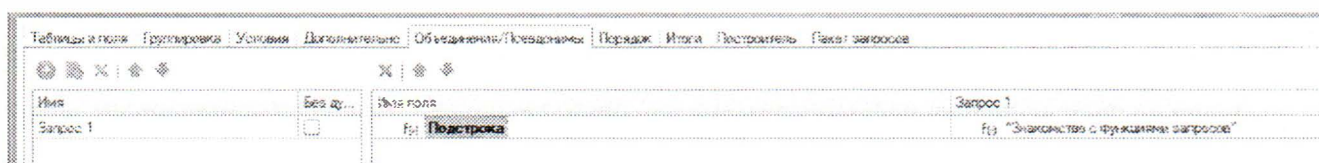


Рисунок 218

Все эти действия мы делаем для знакомства с функциями языка запросов. Сразу скажу, что справку о функциях языка запросов можно найти в справке конфигуратора (Рисунок 219).

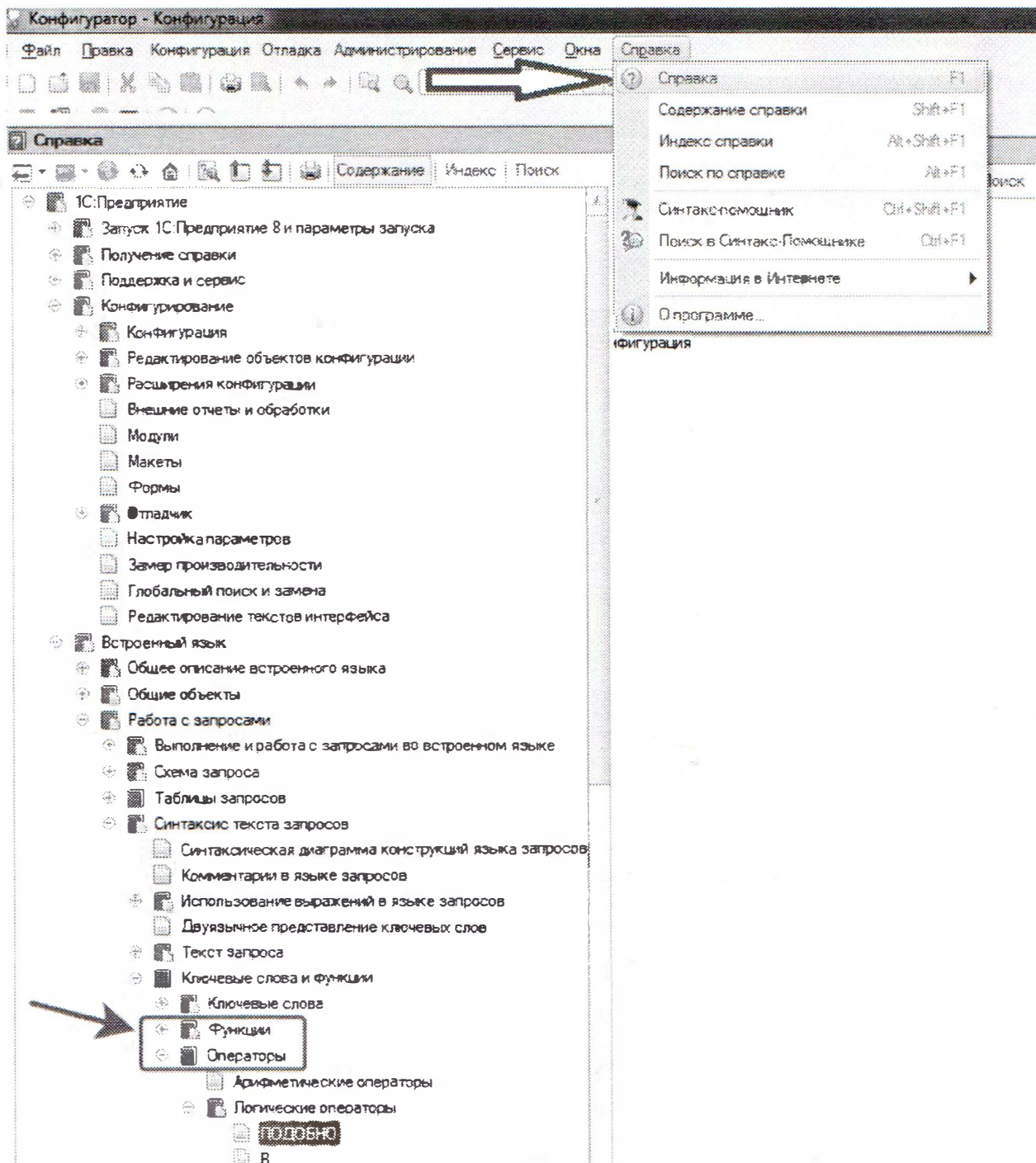


Рисунок 220

Итак, первая функция, с которой мы познакомимся, называется «Подстрока». Она работает со строками и выделяет из строки подстроку. В качестве параметров она принимает исходную строку, начальную позицию и количество символов в виде чисел. То есть мы указываем, с какой позиции (отсчет начинается с единицы) и сколько символов нужно получить. На выходе мы получаем также строку, но как-бы обрезанную этой функцией.

Давайте воспользуемся рассматриваемой функцией для обработки строки. Выделим поле, нажмем на кнопку «Изменить», в правой части окна найдем функцию «Подстрока», перетащим ее в поле редактирования выражения, нашу строку, при этом, переместим на место первого параметра, начальную позицию зададим числом 14, а длину – числом 9 (Рисунок 221).

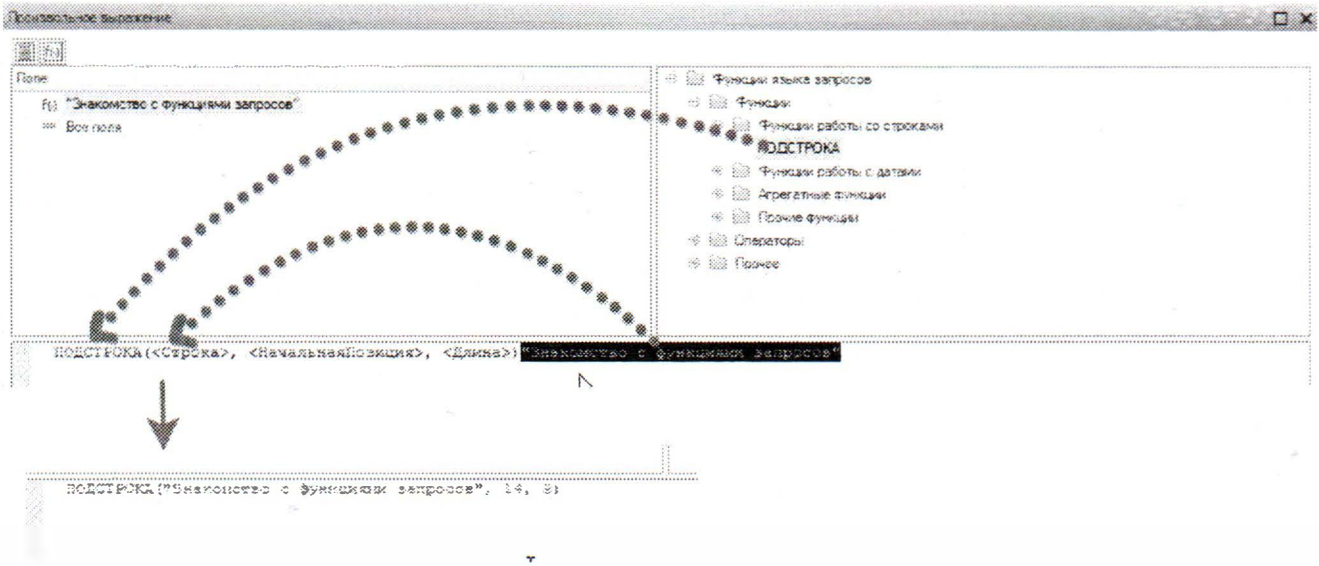


Рисунок 221

Выполним запрос и посмотрим, что получилось (Рисунок 222).

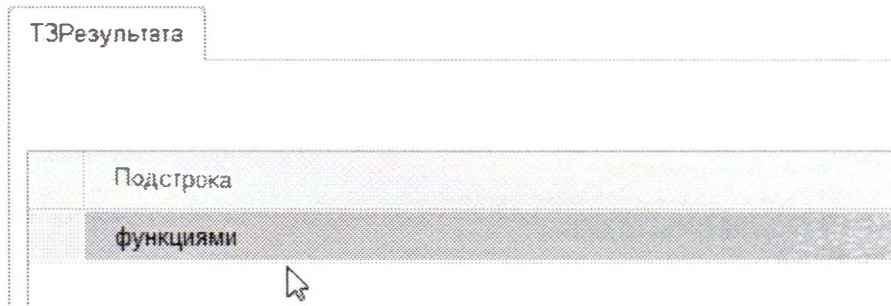


Рисунок 222

Функция «Подстрока» взяла нашу исходную строку и, начиная с 14-го символа, вернула нам 9 последующих символов. Получилась строка «функциями». Это не единственное действие, которое мы можем выполнить со строками. Также можем их «склеивать», что, по-научному, называется конкатенация или сложение строк. Для склеивания мы используем оператор «+», который также является частью языка запросов. Далее добавим к нашей строке символ косой черты (Рисунок 223-224).



Рисунок 223

Текст запроса:

```
ВЫВЕРАТЬ  
ПОДСТРОКА("Знакомство с функциями языка запросов", 14, 9) + "/" КАК Подстрока
```

Рисунок 224

Выполним запрос и посмотрим на результат (Рисунок 225).

ТЗРезультата

Подстрока
функциями /

Рисунок 225

Получилась такая итоговая строка.

Теперь выберем справочник номенклатура, из него возьмем поле «Код» и сложим с полем «Наименование», разделив их все той же косой чертой (Рисунок 226).

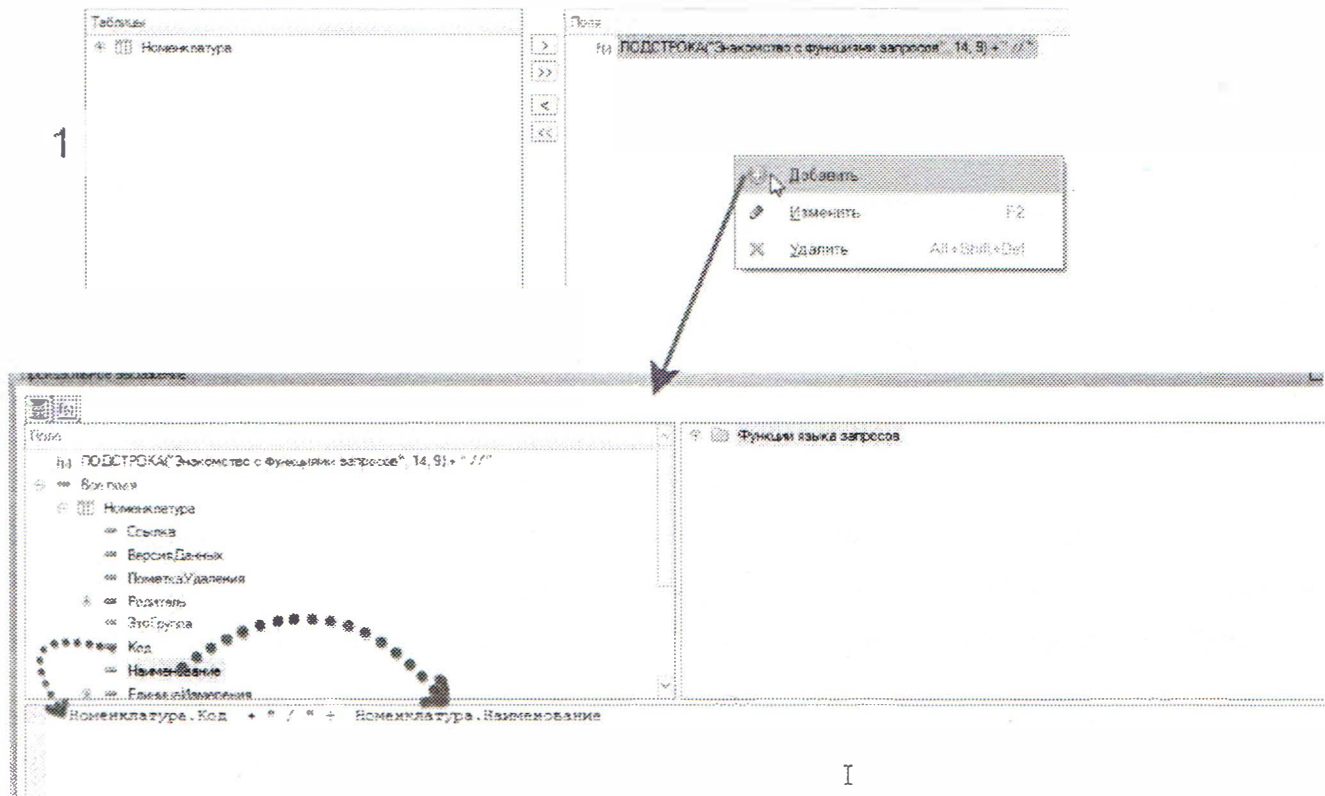


Рисунок 226

Вот что у нас получилось (Рисунок 227).

Подстрока	Поле1
функциями //	00000020 / Кухонные электроприборы
функциями //	00000021 / Комбайн кухонный VINATONE FP 67
функциями //	00000022 / Ксфеварка BRAUN KF22R

Рисунок 227

Этот результат уже может иметь практическое применение для формирования строк в каком-либо отчете или документе.

3.5 Функции языка. Работа с датами.

В языке запросов поддерживается целый ряд функций для работы с датами (Рисунок 228).

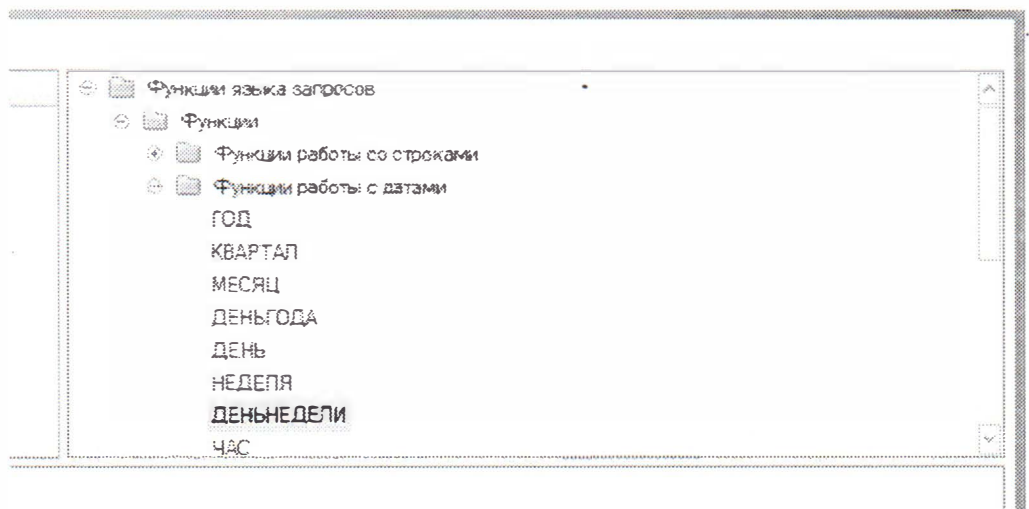


Рисунок 228

Для начала рассмотрим функции: «Год», «Месяц», «День года». В качестве параметра мы передадим в функции параметр запроса. Выглядеть это будет так (Рисунок 229-231).

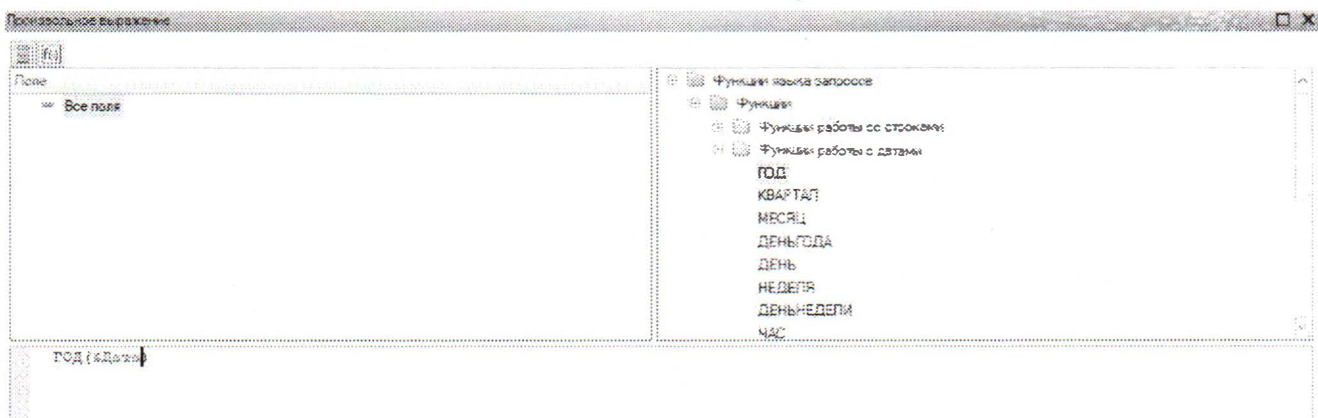


Рисунок 229

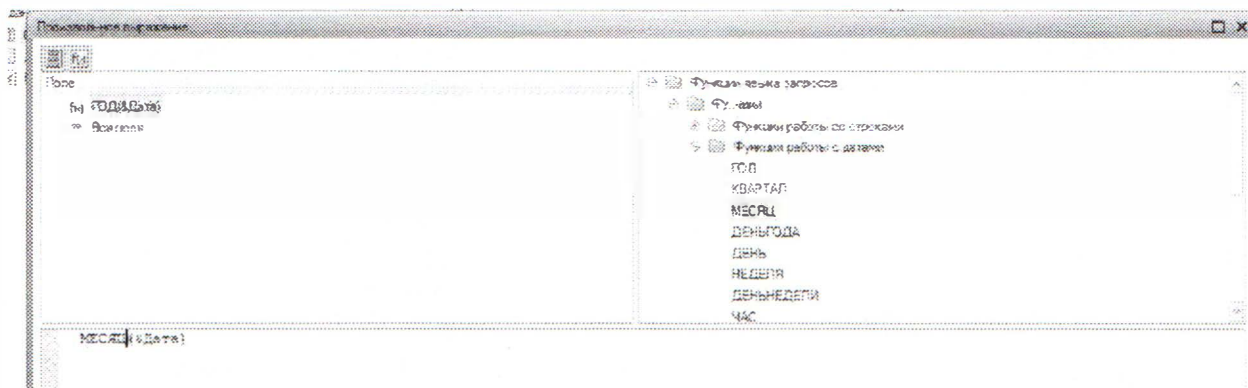


Рисунок 230

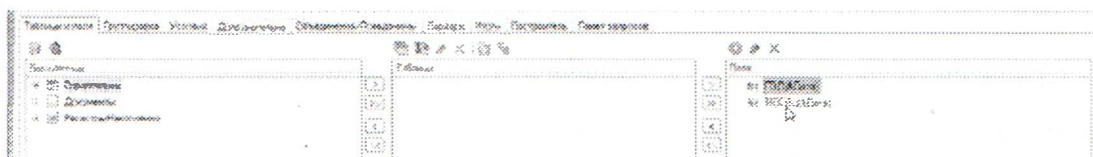


Рисунок 231

Для функции «День года» порядок действий аналогичный. Зададим параметр «Дата» датой 31.01.2014 и выполним запрос.

Результат запроса выглядит следующим образом (Рисунок 232).

ТЗРезультата			
год	месяц	день года	Еще ▾
2014	1	31	

Рисунок 232

Думаю, результат такого запроса, и работа функций не требуют каких-либо особых пояснений, кроме того, что получаемые значения являются числами. Функции: «Минута», «Секунда», «Квартал», «День», «Неделя», «Минута», «Секунда» не должны вызвать вопросов, так как работают аналогично. В качестве самостоятельной работы можете поэкспериментировать с ними, посмотрев, что именно они возвращают.

Функции «Начало периода» и «Конец периода» более интересны, так как приводят дату к началу или концу указанного периода. Указанным периодом может выступать день, месяц, неделя, год... Для указания периода, к началу или к концу которого требуется привести, используются зарезервированные слова языка запросов: день, неделя, месяц, год и т.д.

Мы добавим еще одно поле в наш запрос. Выберем функцию «Начало периода», передадим ей для преобразования параметр запроса &Дата и периодом укажем день (Рисунок 233).

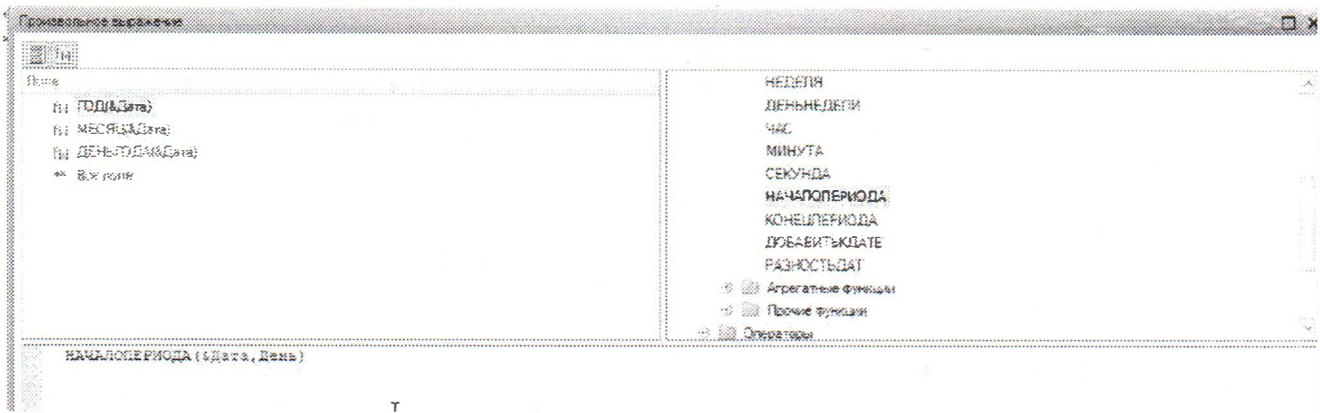


Рисунок 233

Добавим еще одно поле и поступим с ним так же, как и с предыдущим, только в качестве периода укажем месяц (Рисунок 234).

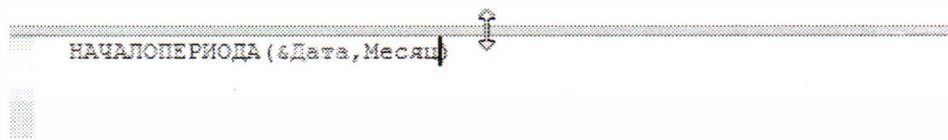


Рисунок 234

В следующем поле выберем функцию «Конец периода» и в качестве периода укажем день (Рисунок 235).

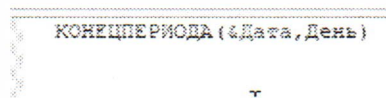


Рисунок 235

Зададим информативные псевдонимы для наших полей (Рисунок 236).

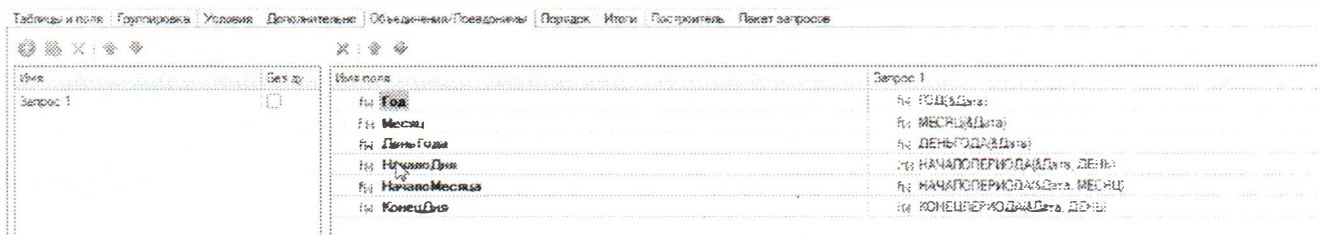


Рисунок 236

Установим в качестве параметра дату 05.02.2014 для большей наглядности. Далее выполним запрос (Рисунок 237).

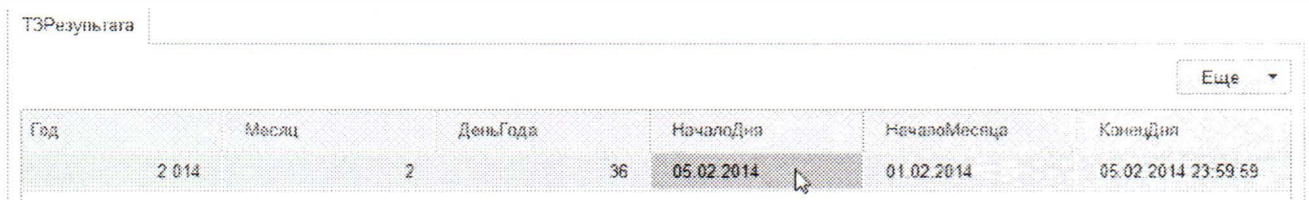


Рисунок 237

В полях «НачалоДня» и «НачалоМесяца» не отображаются часы, минуты и секунды, так как они равны 00:00:00, но фактически они там присутствуют. Вообще, дата в 1С всегда содержит часы, минуты и секунды, только отображаются они не всегда.

Рассмотрим функцию «Добавить к дате». Она принимает три параметра: саму дату, тип и количество. Тип – это тип даты, который будет добавлен. Он может принимать значения: день, месяц, год, минута, секунда, неделя. То есть мы указываем, что будем добавлять: месяцы или дни, секунды или минуты и т.д.

Мы добавим к дате 20 дней, а в новом поле – 3 месяца (Рисунок 238-239).

```
ДОБАВИТЬКДАТЕ (&Дата, День, 20)
```

Рисунок 238

ДОБАВИТЬКДАТЕ (&Дата, Месяц, 3)

Так выглядят все поля нашего запроса на вкладке Таблицы и поля

Поля

- Год(&Дата)
- Месяц(&Дата)
- ДеньГода(&Дата)
- НачалоПериода(&Дата, День)
- НачалоПериода(&Дата, Месяц)
- КонецПериода(&Дата, День)
- ДОБАВИТЬКДАТЕ(&Дата, День, 20)
- ДОБАВИТЬКДАТЕ(&Дата, Месяц, 3)

Зададим псевдонимы

Имя поля	Запрос 1
Год	Год(&Дата)
Месяц	Месяц(&Дата)
ДеньГода	ДеньГода(&Дата)
НачалоДня	НачалоПериода(&Дата, День)
НачалоМесяца	НачалоПериода(&Дата, Месяц)
КонецДня	КонецПериода(&Дата, День)
Добавить20Дней	ДОБАВИТЬКДАТЕ(&Дата, День, 20)
Добавить3Месяца	ДОБАВИТЬКДАТЕ(&Дата, Месяц, 3)

Рисунок 239

Выполним запрос и посмотрим на результат (Рисунок 240).

ТЗРезультата

Еще

ДеньГода	НачалоДня	НачалоМесяца	КонецДня	Добавить20Дней	Добавить3Месяца
36	05.02.2014	01.02.2014	05.02.2014 23:59:59	25.02.2014 0:00:00	05.05.2014 0:00:00

Рисунок 240

Ничего неожиданного.

Рассмотрим функцию «Дата время». Она хоть и располагается в разделе прочее, тем не менее, имеет отношение к нашей теме. Эта функция позволяет задать произвольную дату.

Для этого ей необходимо передать в качестве параметров числовые значения года, месяца, дня, часа, минуты, секунды. Мы сделаем это с помощью литералов (о том, что такое литералы, посмотреть можно в первом уроке) (Рисунок 241).

ДАТАВРЕМЯ (2014, 6, 3, 2, 1, 45)

Рисунок 241

Часы, минуты и секунды можно не указывать, тогда они примут значение 00. Зададим псевдоним для поля «ДатаВремя_2014_6_3_2_1_45». Выполним запрос (Рисунок 242).

ТЗРезультата

НачалоДня	НачалоМесяца	КонецДня	Добавить2Дней	Добавить3Месяца	ДатаВремя_2014_6_3_2_1_45
05.02.2014	01.02.2014	05.02.2014 23:59:59	25.02.2014 0:00:00	05.05.2014 0:00:00	03.06.2014 02:01:45

Рисунок 242

Результат предскажем – мы получили дату 03.06.2014 02:01:45.

Теперь изучим, как работает функция «Разность дат». Она принимает в качестве аргументов две даты, между которыми надо вычислить разность, а также третий аргумент «Тип», который, как в случае с началом и концом периода, указывает, в каких единицах будет вычислена разность. Будут ли это дни или недели, или минуты. Мы передадим в качестве первого аргумента параметр запроса, в качестве второго выберем поле, где устанавливали дату и время при помощи функции «Дата время» и тип выберем «День» (Рисунок 243).

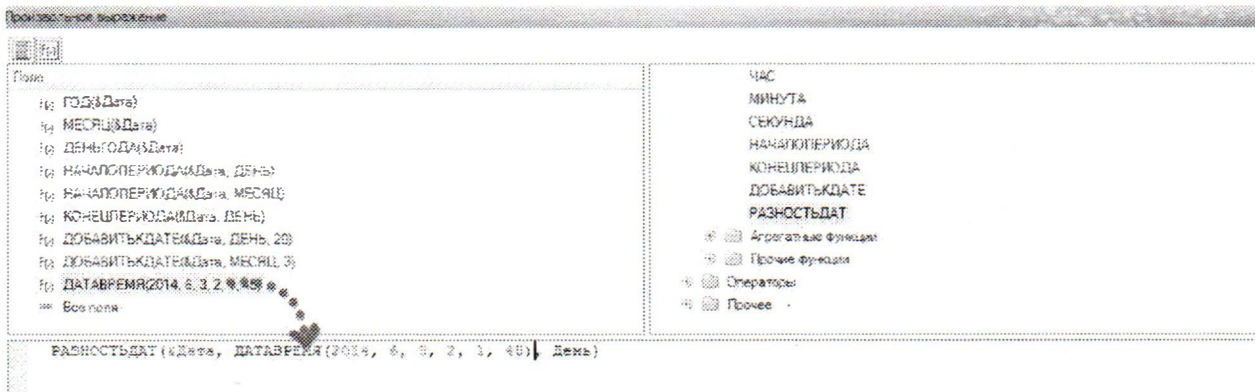


Рисунок 243

Смотрите, какой финт мы проделали с вами. Мы в качестве аргумента функции передали функцию. И в этом нет ничего удивительного. Ведь функция «Дата время» отработав, подставит вместо себя дату и все будет выполнено далее. Зададим псевдоним для этого поля «РазностьДат» и выполним запрос (Рисунок 244).

НачалоМесяца	КонецДня	Добавить20Дней	Добавить3Месяца	ДатаВремя_2014_	РазностьДат
01.02.2014	05.02.2014 23:59:59	25.02.2014 0:00:00	05.05.2014 0:00:00	03.06.2014 2:01:45	118

Рисунок 244

Получили положительное число, то есть, меньшая дата должна передаваться первой, большая – второй. Мы получили разницу в 118 дней, так как задавали разность в днях. Если же мы в аргумент «Тип» напишем месяц, то получим разность в 4 месяца. Предлагаю это проверить самостоятельно.

3.6 Функции языка. Агрегатные функции.

С агрегатными функциями мы сталкивались, когда изучали группировки и итоги, в частях 3 и 4 2-й главы. Сейчас мы рассмотрим подробнее. Напомню, что агрегатные функции – это такие функции, которые принимают на вход много различных значений и выдают одно результирующее значение, например, сумму всех элементов, количество элементов, минимум, максимум и т.д.

Возьмем реальную таблицу «Товары на складах». Выберем из нее поля «Склад» и «Количество». Количество выберем шесть раз. Зачем, поймете чуть позже (Рисунок 245).

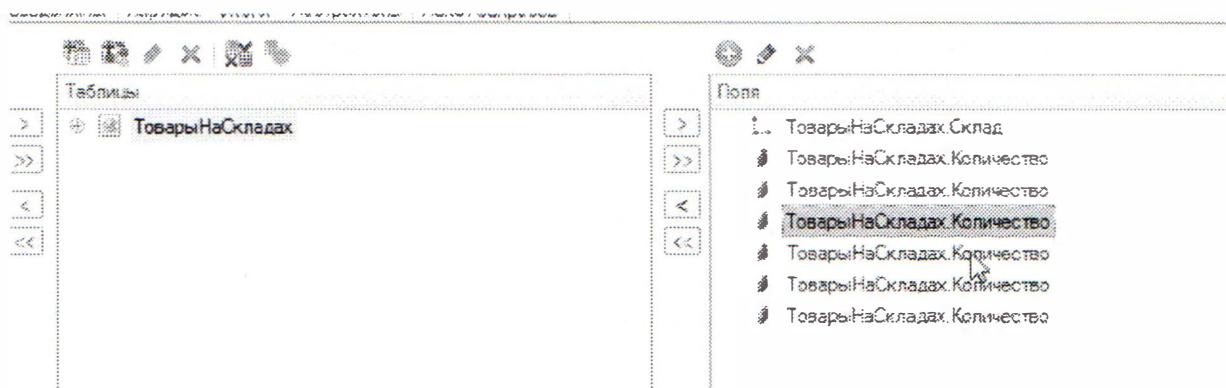


Рисунок 245

А вот на закладке «Группировка» группируем по складу, а для полей количество устанавливаем все возможные агрегатные функции (Рисунок 246).

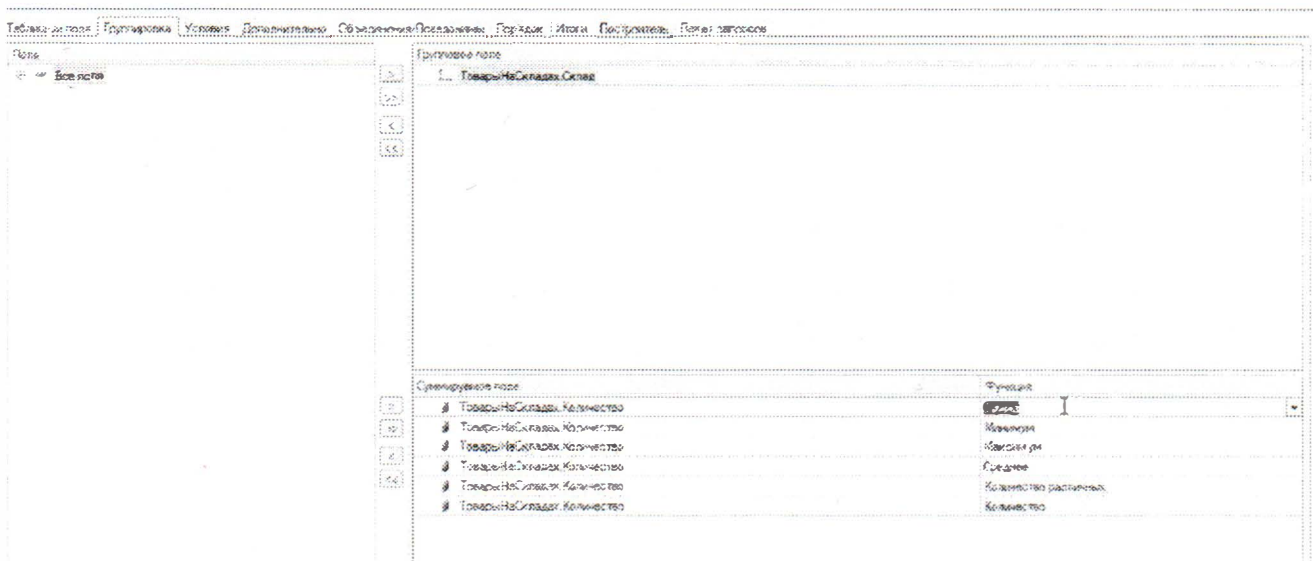


Рисунок 246

Дадим нашим полям соответствующие псевдонимы (Рисунок 247).

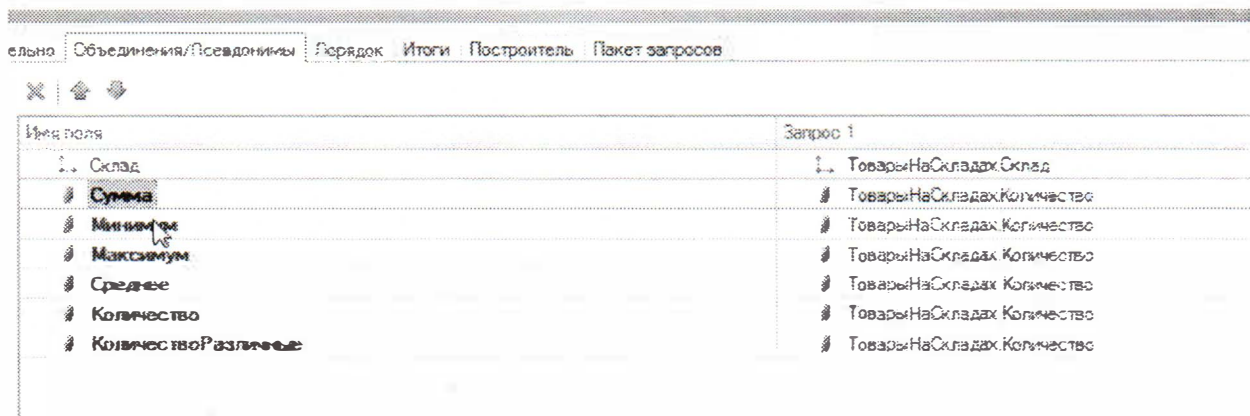


Рисунок 247

Выполним запрос и посмотрим на результат (Рисунок 248).

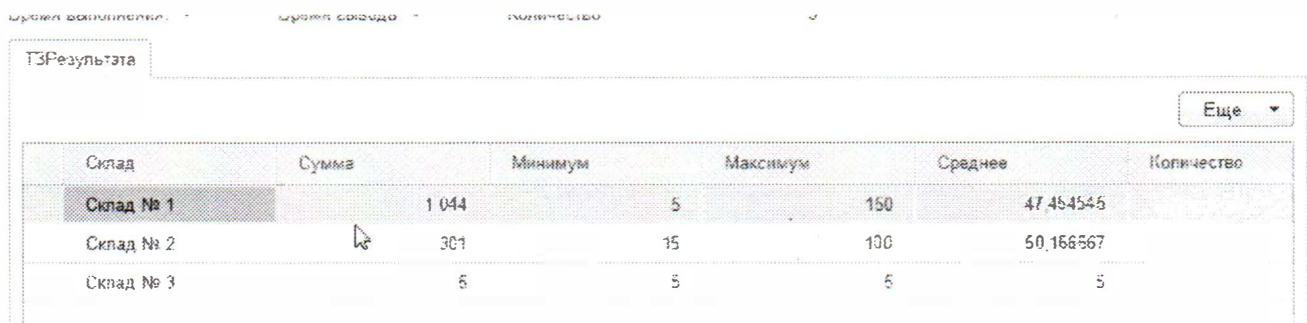


Рисунок 248

По складу №1 мы видим некие цифры, которые мы можем проверить. Через главное меню выберем пункт «Все функции» и выберем регистр «Товары на складах» (Рисунок 249).

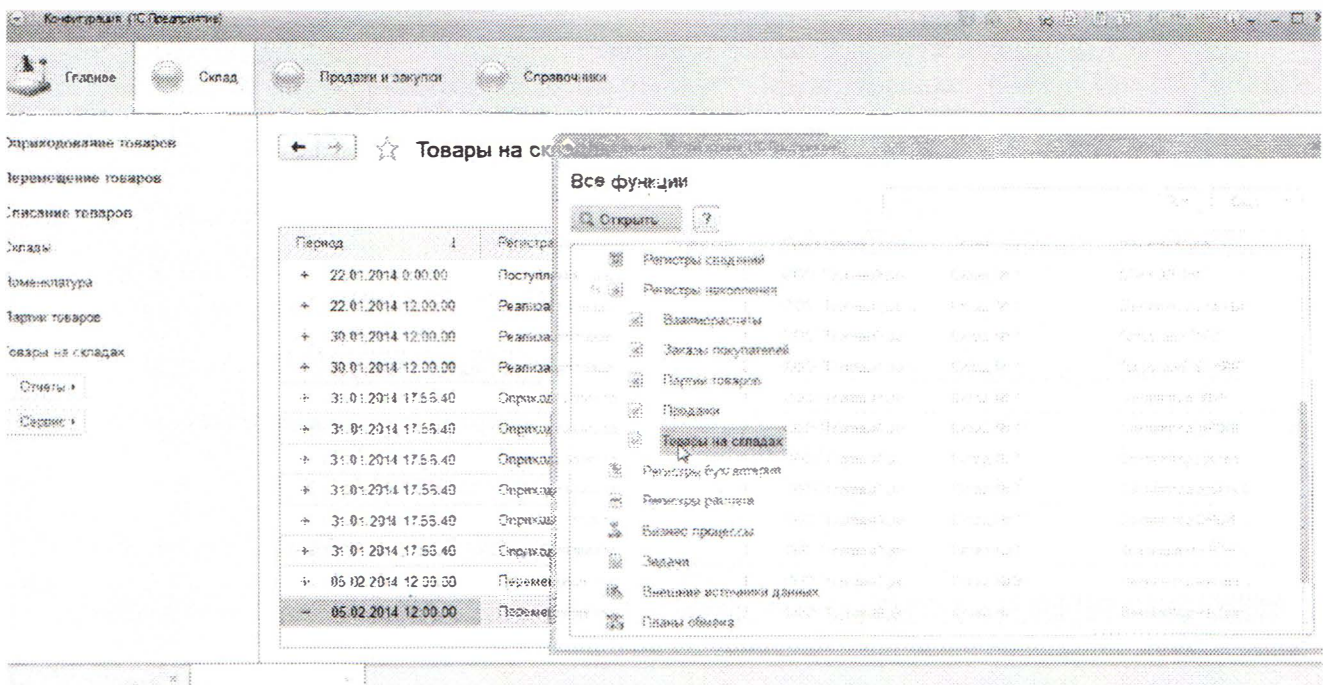


Рисунок 249

Установим отбор строк по складу №1 (Рисунок 250).

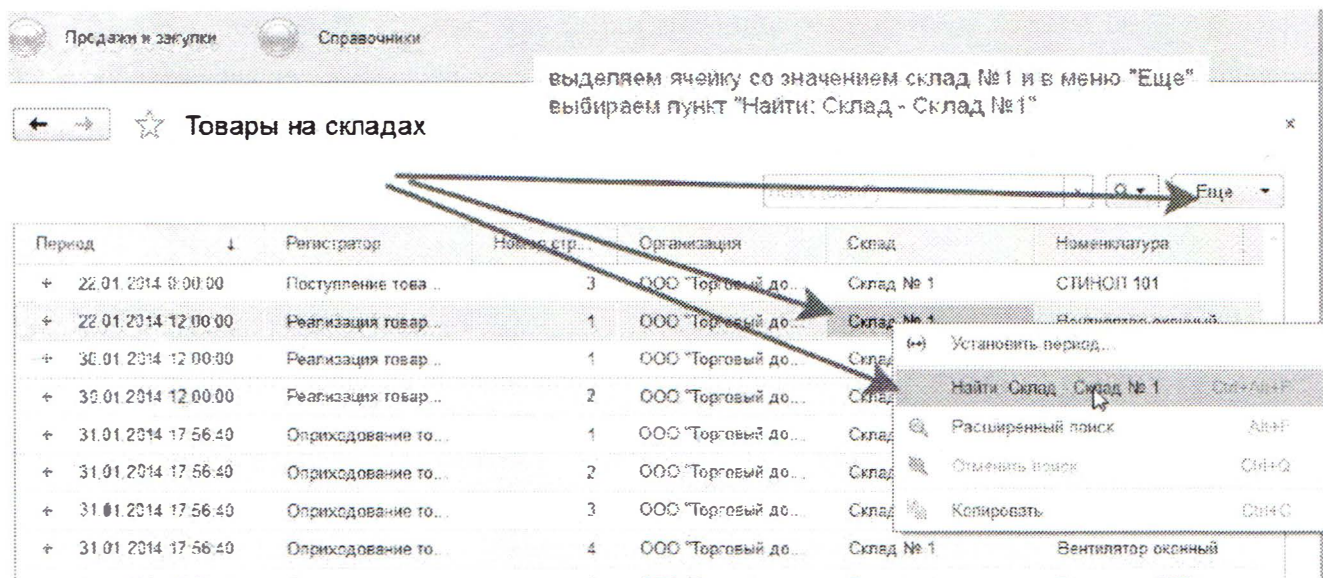


Рисунок 250

Просматриваем все строки, которых, к счастью, не так уж и много. Можно визуально увидеть и минимум, и максимум, которые равны 5 и 150 соответственно. Также мы можем посчитать количество строк и убедиться, что их именно 22, а различных строк, то есть строк с различным количеством, 13. Можем посчитать сумму и убедиться, что она равна 1044.

Вот так просто и наглядно мы посмотрели, как работают агрегатные функции.

3.7 Функции языка. Ссылки и типы.

Рассмотрим ряд функций относящихся к ссылкам и типам. Запросы в этой части будут достаточно простыми, поэтому на них детально останавливаться не будем. При необходимости можете посмотреть их в базе для выполнения домашнего задания. Первой рассмотрим функцию «ЕстьNULL». Передадим ей в первом случае «NULL», а втором – число 10 (Рисунок 251).

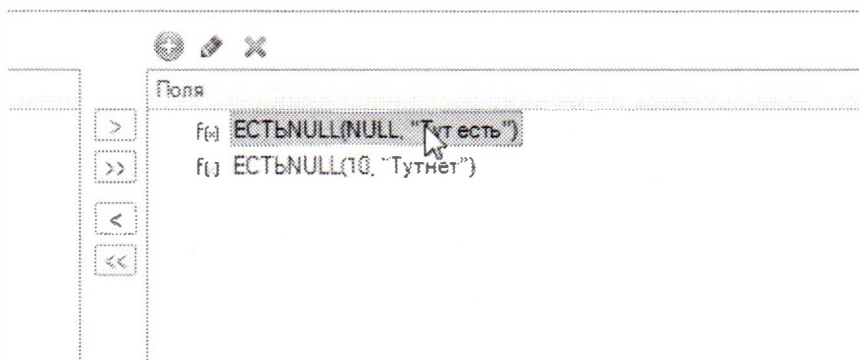


Рисунок 251

В первом случае «NULL» передается при помощи одноименного литерала, поэтому функция должна вывести значение «Тут есть», во втором случае должно быть выведено число 10, так как он не является «NULL» (Рисунок 252).

Время выполнения: - Время вывода: - Количество: 1

ТЗРезультата		Еще ▾
ЕстьNULL	НетNULL	
Тут есть		10

Рисунок 252

Теперь рассмотрим функции «Представление» и «Представление ссылки». Функция «Представление» выводит стандартное представление объекта в виде строки, а функция «Представление ссылки» делает это только для ссылочных типов данных, то есть только для ссылок.

Давайте рассмотрим на примере, в чем разница (Рисунок 253).

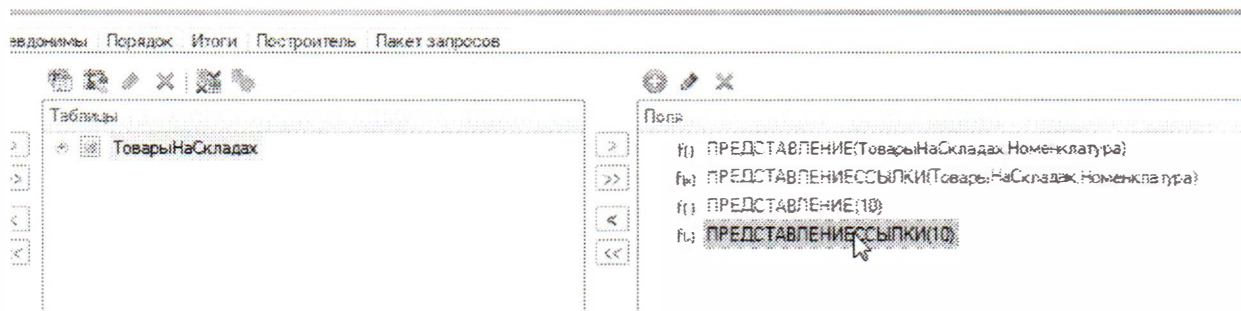


Рисунок 253

В первых двух случаях (для номенклатуры) результат будет одинаковый. И представление, и представление ссылки выведут представление номенклатуры в виде наименования, а это, в свою очередь, задается в конфигураторе. А вот в случае с числом 10, будет разница: первом случае будет выведено представление числа 10 в виде строки, то есть строка «10», во втором – будет выведено число 10, то есть для нессылочного типа данных эта функция не работает.

Следующие функции чаще используются в паре («Тип значения» и «Тип»). Их используют, как правило, для сравнения в условиях, чтобы проверить тип значения (Рисунок 254).

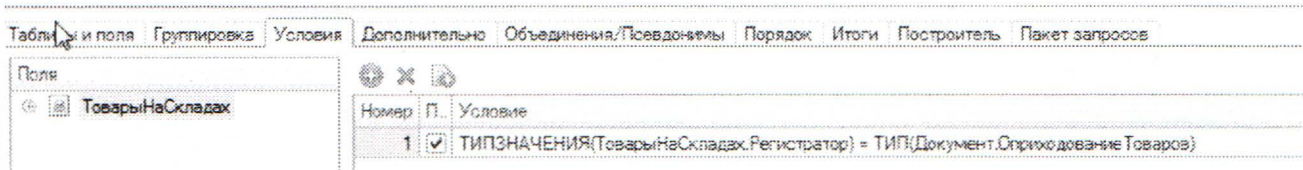


Рисунок 254

В данном случае в запросе из регистра «Товары на складах» мы выбрали поле регистратор и поставили условие, что типом значения регистратора должен быть документ «Оприходование товаров». Если мы выполним такой запрос, то получим список регистраторов только типа документ «Оприходование товаров» (Рисунок 255).



Рисунок 255

Хотя регистратором может быть и реализация товаров.

Следующая функция – «Значение». Эта функция позволяет получать в запросе predetermined значения или пустые ссылки. Это могут predetermined значения справочников или значения перечислений. Так, например, в нашей конфигурации есть predetermined единица измерения – штука. В этом мы можем убедиться, открыв конфигуратор и посмотрев свойства справочника (Рисунок 256).

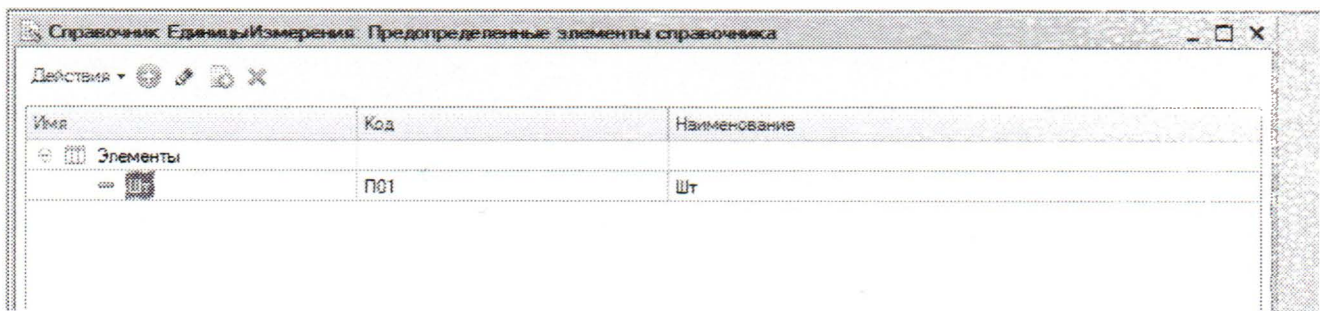


Рисунок 256

К этому значению мы можем обратиться прямо из запроса (Рисунок 257).

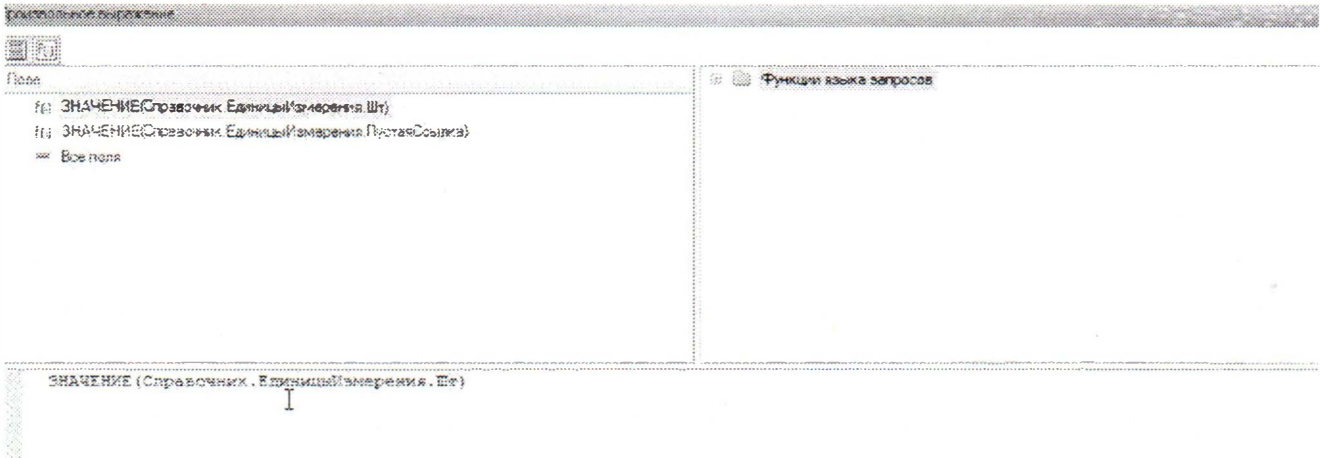


Рисунок 257

Во встроенном языке мы обращаемся к справочнику при помощи слова «Справочники», далее точка и имя справочника. В языке запросов это происходит иначе. И в функции «Тип», и в функции «Значение» мы пишем «Справочник» или «Документ» (в единственном числе). Вторым полем в запросе мы получим пустую ссылку (Рисунок 258).

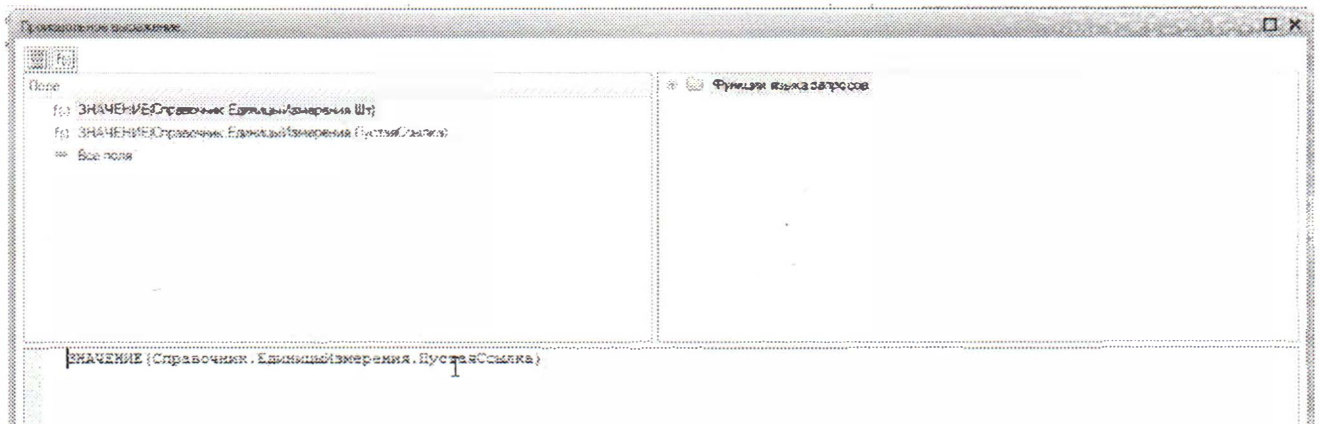


Рисунок 258

Также пустую ссылку можно получить и для любого типа документа. Зачем нужна пустая ссылка и какая от нее польза? В первую очередь, для задания условий. Так, например, мы можем обратиться к справочнику номенклатура и отбросить все элементы, где единицы измерения не заданы, то есть равны пустой ссылке.

Выполним запрос и посмотрим, что получилось (Рисунок 259).

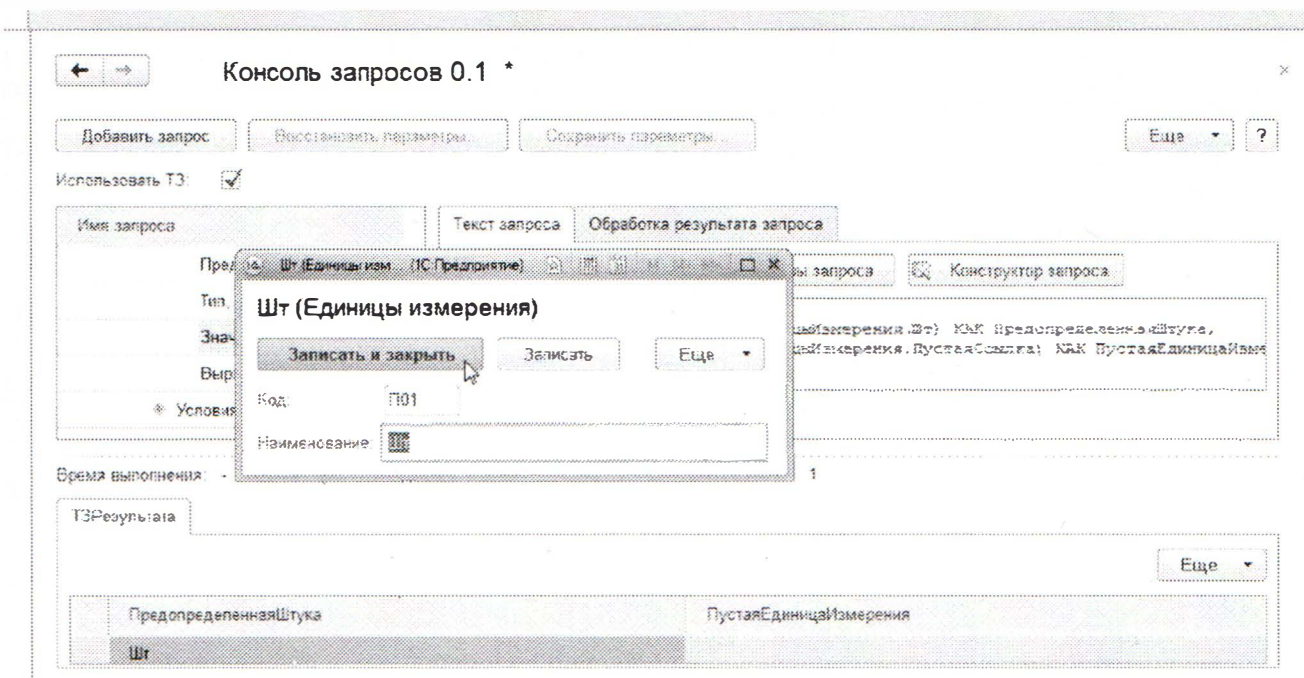


Рисунок 259

Мы видим, что есть предопределенная единица измерения «Штука», она даже открылась при двойном клике по ячейке, а во второй ячейке находится пустая ссылка, но посмотреть на нее мы не сможем.

Функция «Выразить» нужна для того, чтобы значения, имеющие составной тип, были преобразованы к определенному типу, который входит в этот составной тип. Например, регистратор регистра накопления «Товары на складах» имеет составной тип, то есть может быть оприходованием, может быть реализацией и может быть поступлением. Если мы выразим регистратор только как поступление товаров, то остальные типы документов не будут отображены в выборке, хотя строки будут присутствовать, но поле регистратор заполнено в них не будет, так как другие типы документов не могут быть выражены как документ оприходования товаров (Рисунок 260).

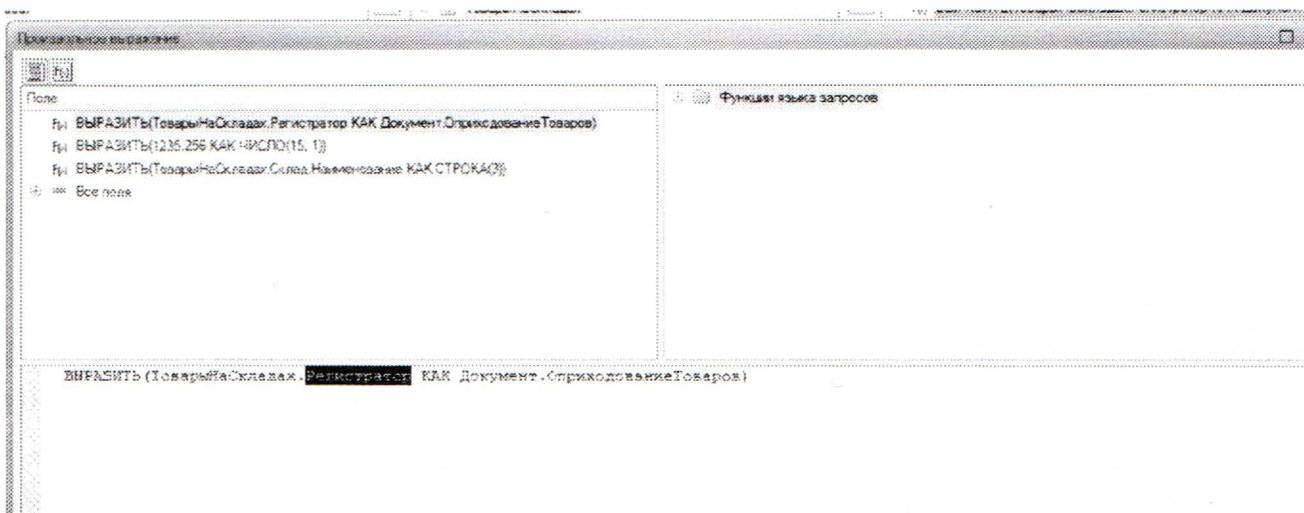


Рисунок 260

Также мы можем выразить число с тремя знаками после запятой как число с одним знаком после запятой (Рисунок 261).

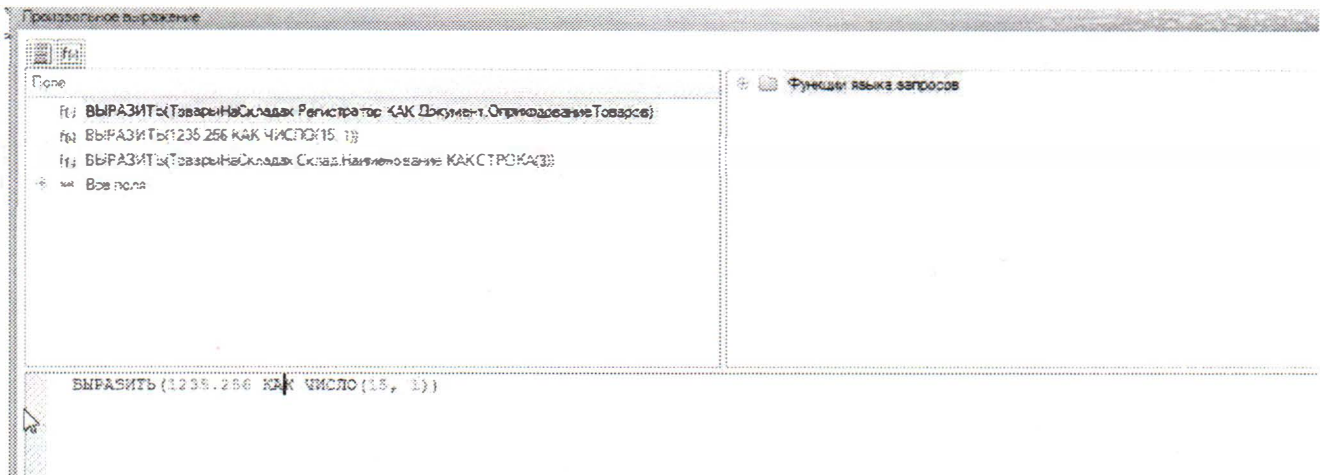


Рисунок 261

Кроме того, мы можем преобразовывать строки в строки заданной длины. Например, наименование в справочнике «Склады» является строкой, но мы его можем выразить строкой с длиной 3. Значит, все, что в наименовании идет после третьего символа, будет обрезано (Рисунок 262-263).

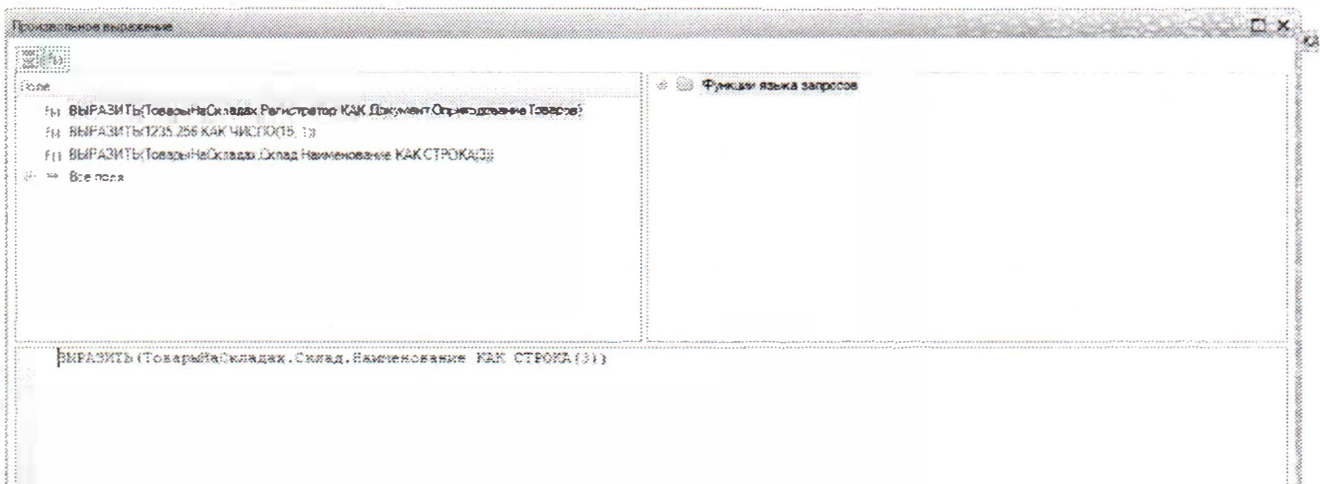


Рисунок 262

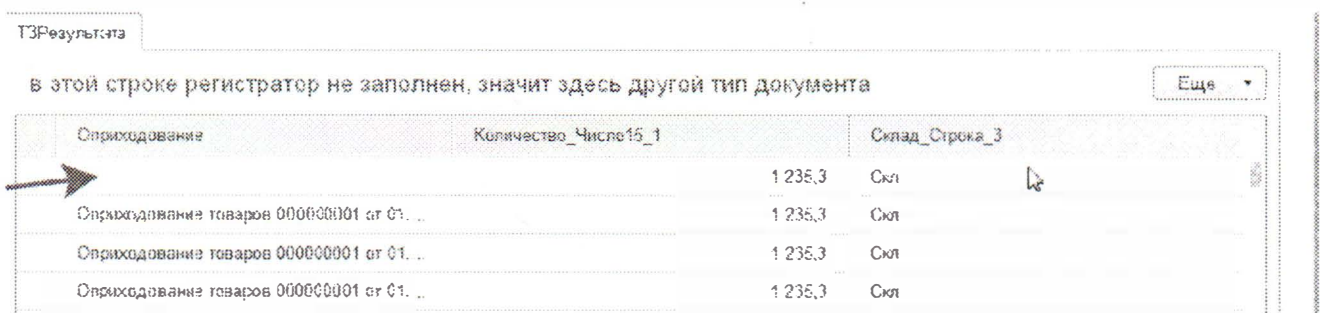


Рисунок 263

Видим, как преобразовалось число. Это своего рода округление средствами языка запросов. Наименование склада обрезано до 3х символов.

Преобразование строк бывает полезно, когда нужно обработать строки неограниченной длины. Зачастую строками неограниченной длины бывает реквизит комментарий в документах.

В то же время, функция «Выразить» обладает и ограничением. Нельзя превратить один тип данных в другой. Например, нельзя превратить строку в число. Если попытаемся это сделать, то получим ошибку (Рисунок 264).

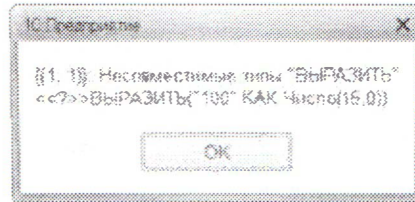


Рисунок 264

3.8 Функции языка. Условие. Ссылка.

Среди условных операторов есть оператор «Ссылка». Он действует также, как проверка на тип значения, то есть тем же самым, что мы рассмотрели в предыдущей части про функции «Тип значения» и «Тип». Тот же принцип действия.

Выберем регистратор из регистра «Товары на складах» (Рисунок 265).

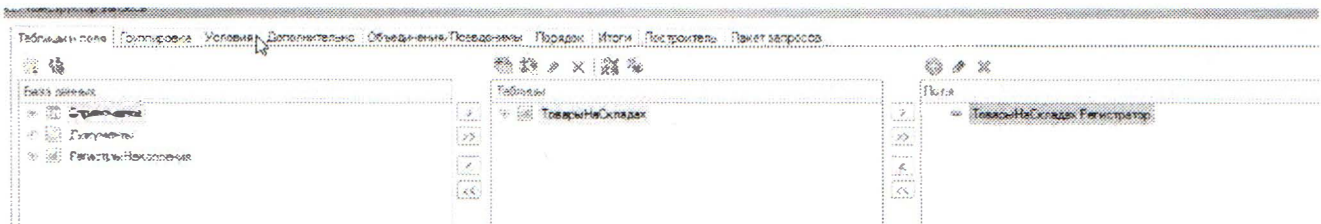


Рисунок 265

И зададим такое условие (Рисунок 266).

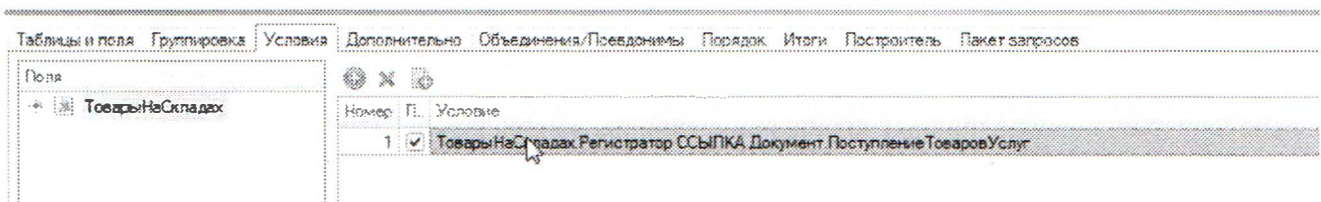


Рисунок 266

То есть условие говорит, что тип значения регистратора должен быть документ «Поступление товаров и услуг».

Выполним запрос и убедимся, что остались только поступления товаров и услуг в выборке (Рисунок 267).

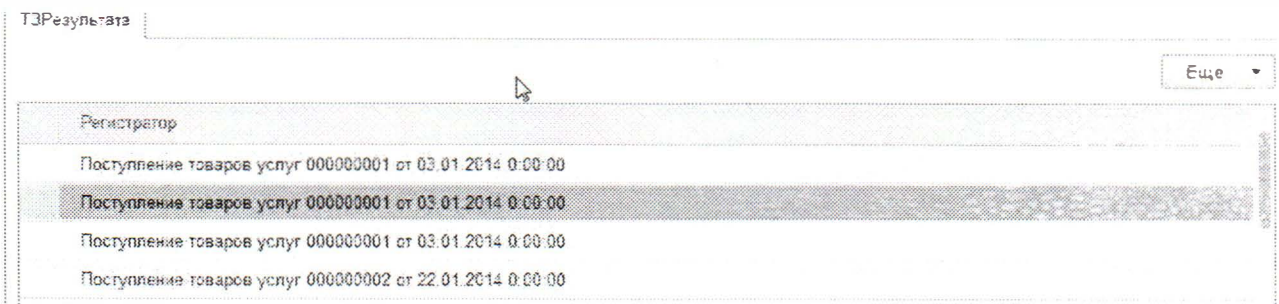


Рисунок 267

3.9 Функции языка. Условия. В, В иерархии.

Оператор «В» проверяет вхождение в некий список, то есть мы можем указать список значений. Если элемент входит в этот список, то условие выполняется. Например, выбираем ссылки из справочника номенклатура и задаем условие – ссылка должна быть в списке номенклатуры (Рисунок 268).

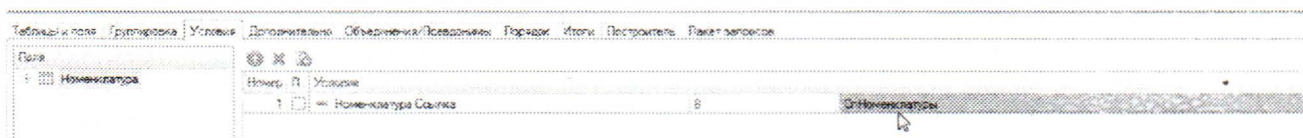


Рисунок 268

«СпНоменклатуры» – это параметр запроса. Прежде чем выполнить запрос установим параметр, при этом указав, что это список (Рисунок 269).

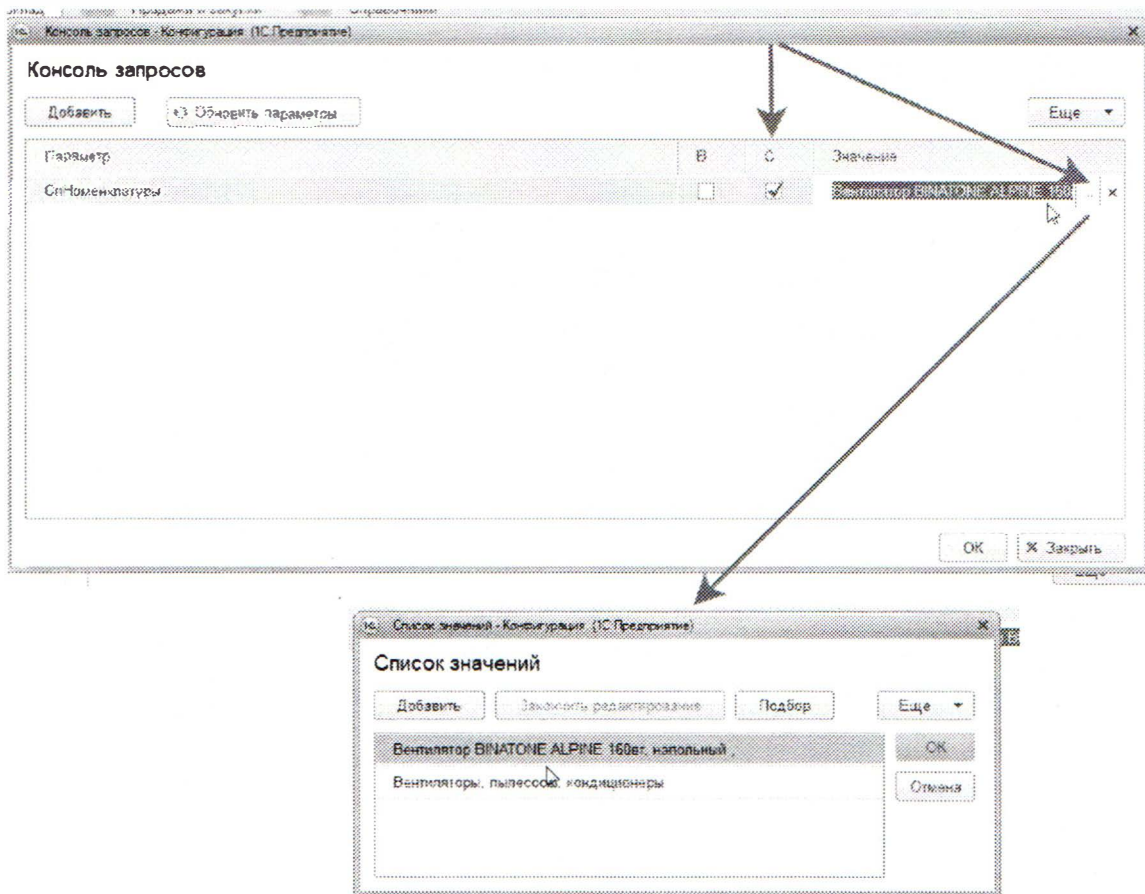


Рисунок 269

Теперь задаем список значений и выполняем запрос (Рисунок 270).

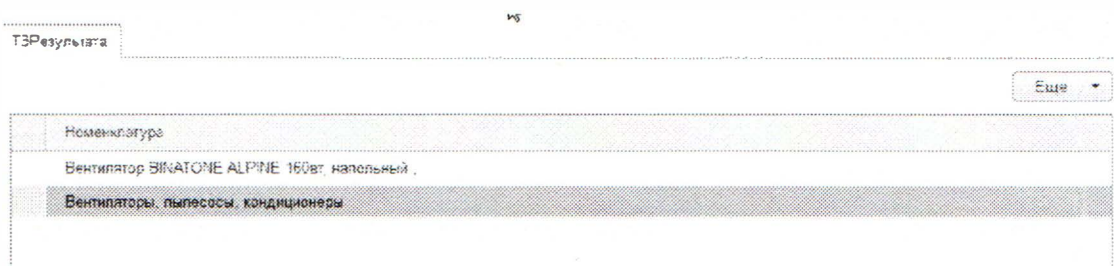


Рисунок 270

Оператор «В иерархии» выводит все содержимое некой группы или иерархического элемента. Ведь в 1С иерархия может быть не только групп, но и элементов, а также групп и элементов. Допустим, мы хотим вывести все содержимое какой-либо папки (группы) справочника «Номенклатура». Вот тут нам и поможет оператор «В иерархии», в котором мы укажем группу в качестве параметра. Будут выведены все элементы этой группы, включая вложенные группы и их содержимое. Запрос используем из предыдущей части, меняем только условие (Рисунок 271).

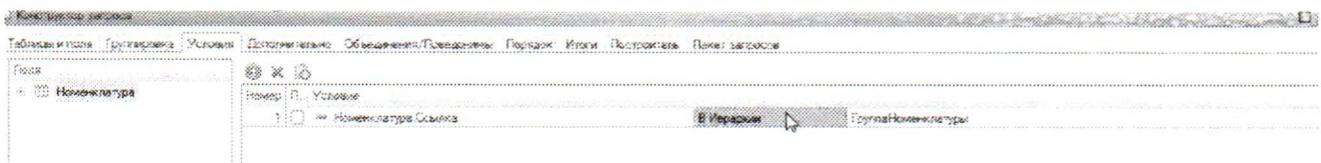


Рисунок 271

Указываем в параметрах группу (Рисунок 272).

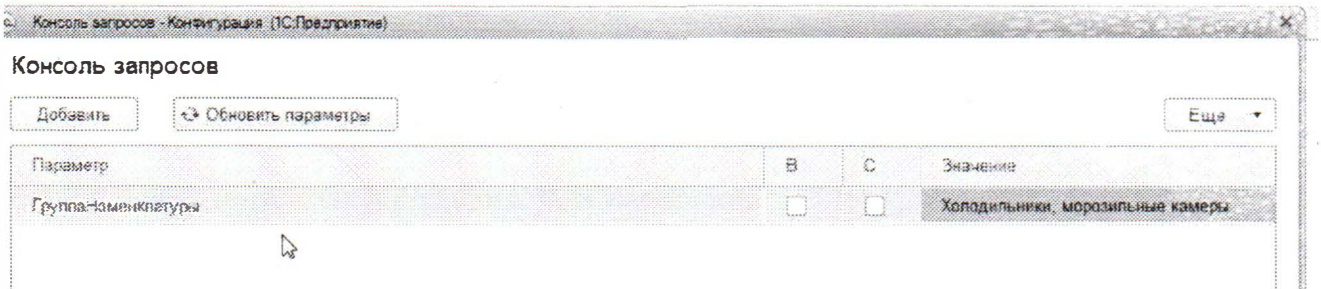


Рисунок 272

Выполняем запрос и видим, что выведено только то, что входит в папку «Холодильник, морозильные камеры» (Рисунок 273).



Рисунок 273

Следует обратить внимание, что если в параметрах передать пустую ссылку, то есть не заполнить параметр, то мы получим весь справочник в результате. Это происходит по причине того, что все папки и элементы верхнего уровня имеют незаполненный реквизит «Родитель». Незаполненный реквизит – пустая ссылка. Таким образом, все содержимое папок верхнего уровня попадает в выборку, как и элементы, находящиеся на верхнем уровне. Попробуйте очистить параметр запроса и выполнить запрос, и вы убедитесь в том, что будет выведен весь справочник.

3.10 Функции языка. Условия. Есть NULL.

Оператор «Есть NULL» (не путать с функцией ЕСТЬNULL) проверяет, равно ли выражение (значение поля) NULL. Дело в том, что в языке запросов NULL – это особый тип данных, который нельзя проверить на равенство оператором «=». Такая операция сравнения всегда будет возвращать значение «Ложь», поэтому существует специальный логический

оператор «Есть NULL». Для проверки на NULL мы передадим значение NULL с помощью литерала, как мы это делали ранее. Наш пример носит чисто учебные цели и в обычной жизни такое выражение бессмысленно. Но, в нашем случае, оно демонстрирует, что оператор «Есть NULL» отлично справляется с проверкой на NULL.

Итак, создадим запрос с одним полем в виде произвольного выражения (Рисунок 274).

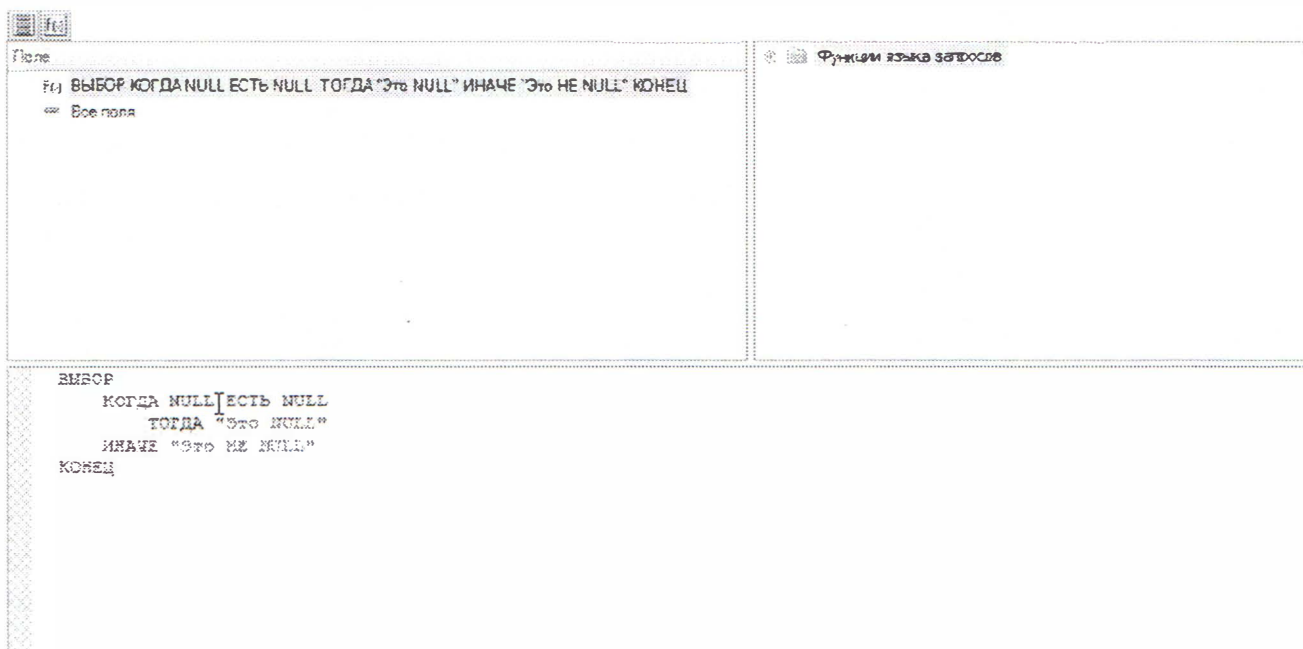


Рисунок 274

Оператор «Есть NULL» проверяет, является ли выражение слева от него значением NULL, если является, то возвращает истину, то есть условие выполнено. Выполним запрос (Рисунок 275).

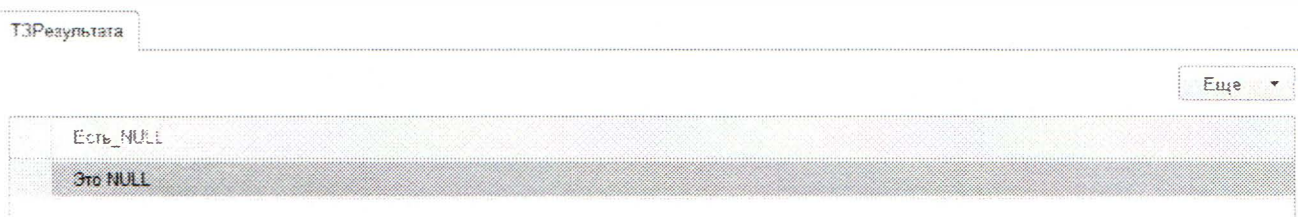


Рисунок 275

Что и требовалось доказать.

3.11 Функции языка. Условия. Выбор.

Оператор «Выбор» позволяет нам осуществлять ветвление при выполнении запроса. Это аналог оператора «Если» в языке программирования 1С. То есть, в зависимости от условий, может быть выбрано (или рассчитано) то или иное значение. Обратимся к виртуальной таблице «ТоварыНаСкладахОбороты», выберем поля «Номенклатура» и «КоличествоОборот». Третье поле будет условием, то есть мы как раз применим наш

оператор «Выбор». Если перетащить оператор из окна функций, то мы увидим шаблон заполнения этого оператора (Рисунок 276).

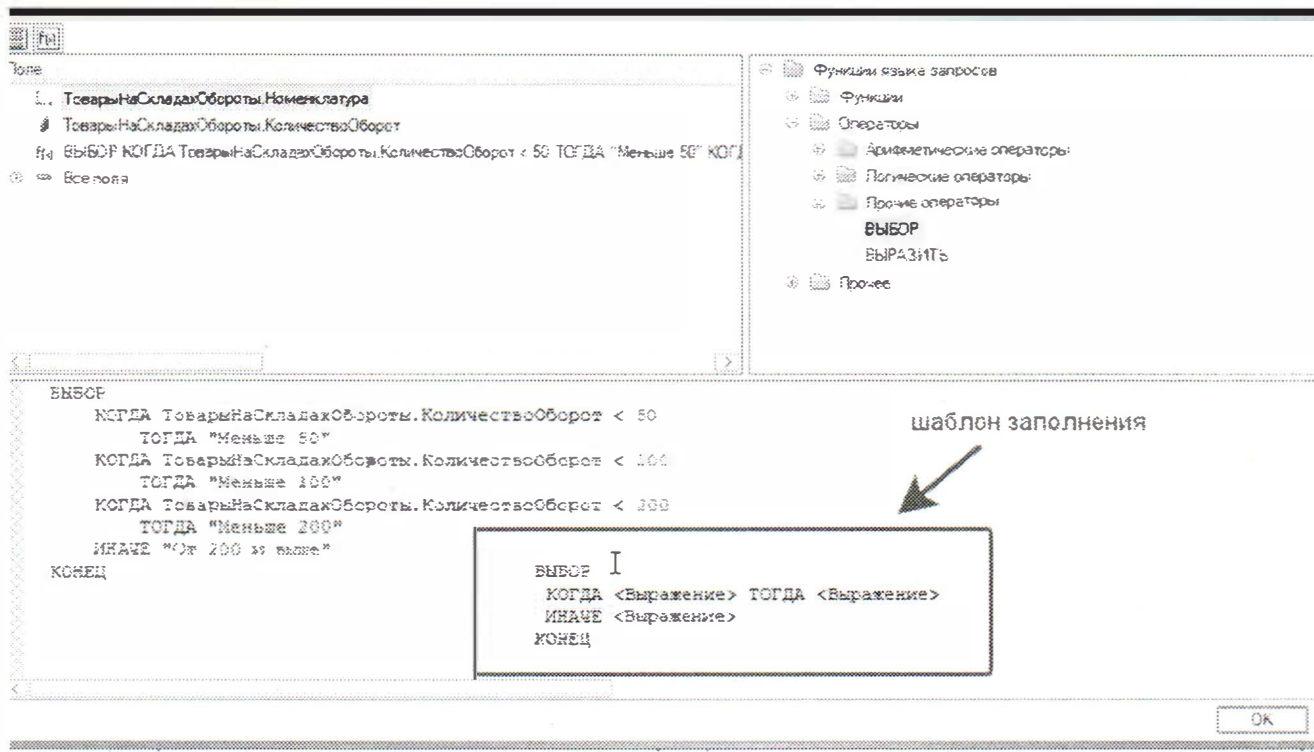


Рисунок 276

Начинается оператор с зарезервированного слова «Выбор». Далее идет условие «Когда <Выражение> Тогда <Выражение> Иначе <Выражение>». Вместо «<Выражение>» после слова «Когда» необходимо вставить логическое выражение. Секций «Когда <Выражение> Тогда <Выражение>» может быть много: условие можно задавать сколь угодно сложным. Заканчивается оператор зарезервированным словом «Конец».

Мы же зададим следующие условия (Рисунок 277).

```
ВЫБОР
    КОГДА ТоварыНаСкладахОбороты.КоличествоОборот < 50
        ТОГДА "Меньше 50"
    КОГДА ТоварыНаСкладахОбороты.КоличествоОборот < 100
        ТОГДА "Меньше 100"
    КОГДА ТоварыНаСкладахОбороты.КоличествоОборот < 200
        ТОГДА "Меньше 200"
    ИНАЧЕ "От 200 и выше"
КОНЕЦ
```

Рисунок 277

Это значит, что когда количество будет меньше 50, будет выведена строка «Меньше 50», когда количество будет меньше ста, будет выведена строка «Меньше 100», когда количество будет меньше 200, то будет выведена строка «Меньше 200». Если же ни одно из условий не будет выполнено, то будет выведена строка «От 200 и выше». Необходимо обратить внимание на важный нюанс. Допустим, количество будет равно 3. Тогда оно

удовлетворяет и первому условию, и второму, и третьему! Какое же из них будет выполнено? Будет выполнено первое из этих условий, остальные не будут проверяться. Поэтому, чтобы такое условие работало правильно, я разместил их в порядке возрастания значений. Если бы первым располагалось условие «Когда ТоварыНаСкладахОбороты.КоличествоОборот < 100», то в рассматриваемом нами случае, со значением количества 3, это условие было бы проверено первым, и была бы выведена строка «Меньше 100». Чтобы не получать таких неприятных сюрпризов помните об этой особенности и задавайте условия корректно. Присвоим этому полю псевдоним «Описание». Выполним запрос и посмотрим, правильно ли отработал оператор «Выбор» (Рисунок 278).

Номенклатура	КоличествоОборот	Описание
СТИНОЛ 101	22	Меньше 50
Вентилятор BINATONE ALPINE 160Вт, нап	266	От 200 и выше
Вентилятор JAPONIC (Таёв)	262	От 200 и выше
Вентилятор настольный	264	От 200 и выше
Вентилятор оконный	155	Меньше 200

Рисунок 278

Как видно, ошибок нет: 22 меньше 50, хотя оно и меньше 100 и 200, но выполнилось первое условие.

Помимо обработки значений поле, оператор «Выбор» можно использовать в условиях. Рассмотрим такой пример. Обратимся к той же таблице и выберем организацию, номенклатуру, склад, количество (Рисунок 279).

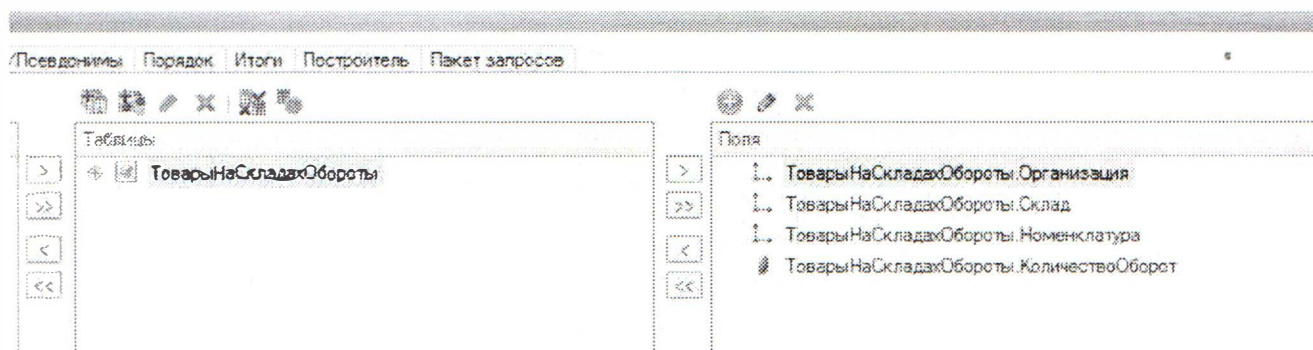


Рисунок 279

Теперь нам необходимо задать условие и задать его нужно в виде произвольного выражения, тогда мы сможем использовать операторы и функции языка запросов при редактировании условия. Условие будет таким: если склад равен складу №2, то выводить только те строки, у которых количество больше 30, для всех остальных складов выводить строки без ограничений. Попробуем это реализовать (Рисунок 280).

1. добавляем условие, нажав сюда

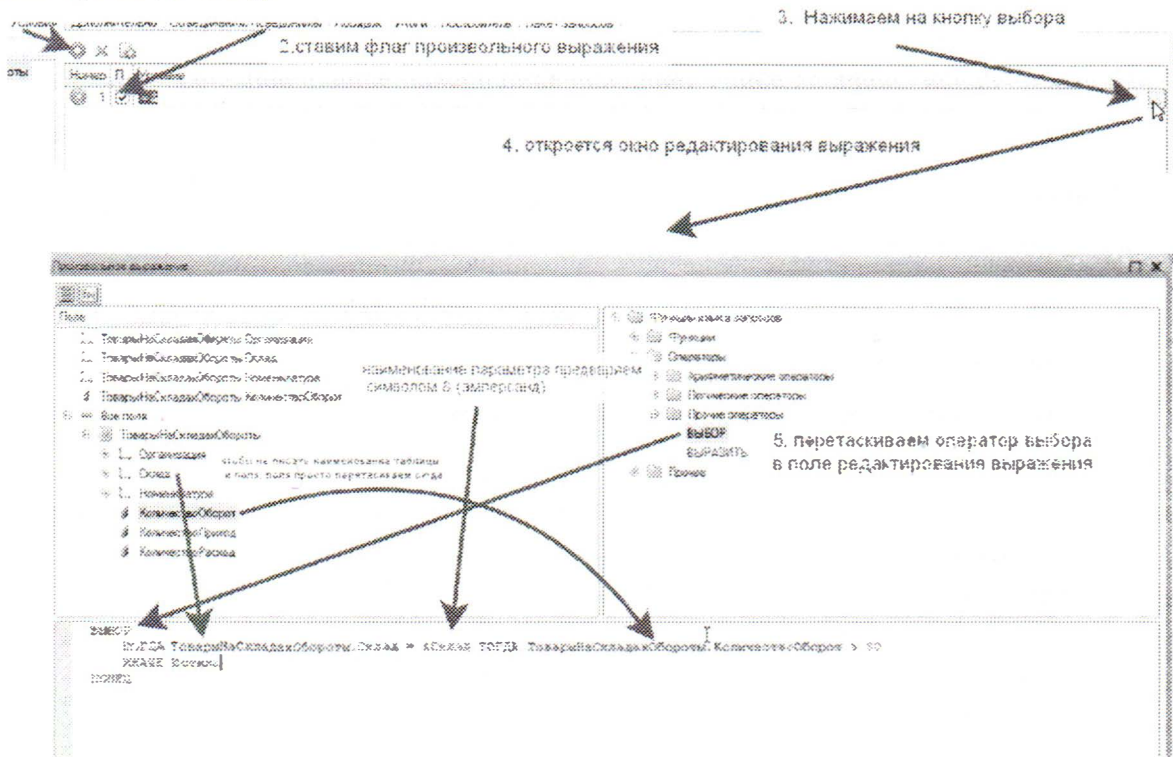


Рисунок 280

Рассмотрим подробнее то, что изображено на рисунке 280.

Мы добавили условие, отметили, что оно произвольное. Открыли окно редактирования выражения. Перетащили оператор «Выбор» в нижнюю часть окна, где выполняется редактирование выражения. Из шаблона удалили «<Выражение>», заполнили условия и выражения.

Как работает условие: в условии задается логическое выражение, результат выполнения которого равен либо «Истина» (условие истинно, выполнено), либо «Ложь» (условие не выполнено). В нашем выражении проверяется условие, другими словами, мы задаем логическое выражение, результатом которого будет либо «Истина», либо «Ложь». Если количество больше 30, значение выражения – «Истина», то есть условие выполнено, строка попадет в выборку, в противном случае, условие не выполнится, и строка в выборку не попадет. Для всех остальных складов, которые не равны параметру «Склад», выражение всегда истинно, то есть попадут все строки без ограничений. Это мы указали в секции «Иначе Истина», то есть если склад не равен складу, указанному в параметре «Склад», то условие всегда будет выполняться (будет истинно).

Зададим параметр и выполним запрос (Рисунок 281-282).

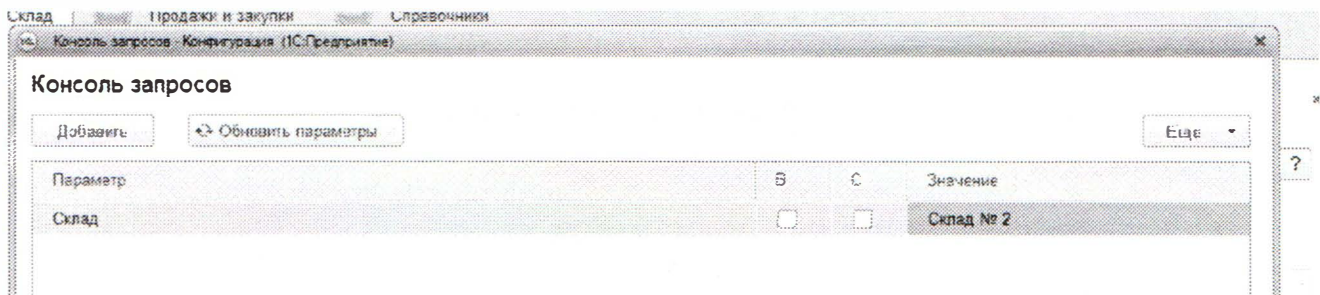


Рисунок 281

ТЗРезультата

Организация	Склад	Номенклатура	КоличествоОборот
ООО "Торговый дом"	Склад № 2	Вентилятор BINATONE ALPIN	94
ООО "Торговый дом"	Склад № 2	Вентилятор JIPONIC (Тайв)	32
ООО "Торговый дом"	Склад № 2	Вентилятор настольный	40
ООО "Торговый дом"	Склад № 3	Вентилятор настольный	5

Рисунок 282

Можете проверить и убедиться, что ни одна строка с количеством меньше 30 для склада №2 не попала в таблицу, а для других складов попали все строки.

3.12 Функции языка. Условия. Подобно.

Рассмотрим оператор «Подобно».

«Подобно» позволяет сравнивать строковые значения с шаблонами. Для задания шаблонов используются специальные символы.

Следующие символы в строке шаблона являются служебными и имеют смысл, отличный от символа строки:

- % (процент): последовательность, содержащая любое количество произвольных символов.

- _ (подчеркивание): один произвольный символ.

- [...] (в квадратных скобках один или несколько символов) – любой одиночный символ из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например a-z, означающие произвольный символ, входящий в диапазон, включая концы диапазона.

- [^...] (в квадратных скобках значок отрицания, за которым следует один или несколько символов) – любой одиночный символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки.

Рассмотрим на примере.

Выберем ссылки из справочника номенклатура, а в качестве условия зададим (Рисунок 283).

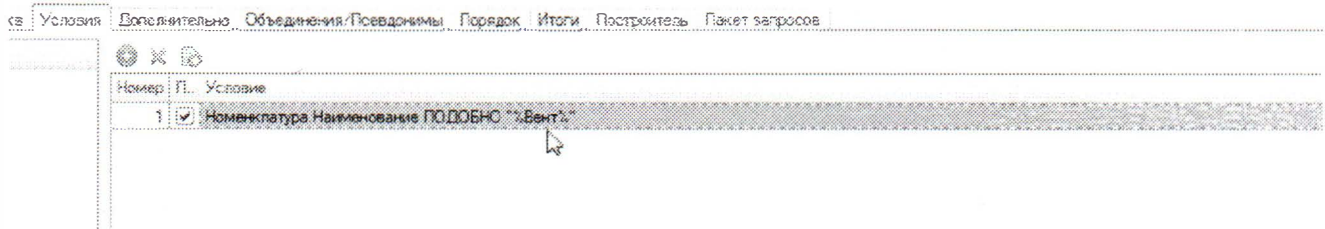


Рисунок 283

Это значит, что в таблицу должна попасть номенклатура, содержащая в своем наименовании подстроку «Вент». При этом сам шаблон мы обрамили кавычками, то есть шаблон задается в виде строки. Символ процента значит любое количество любых символов. Получается, что шаблон значит буквально следующее: любое количество любых символов до, последовательность символов «Вент», затем любое количество символов после.

Выполним запрос (Рисунок 284).

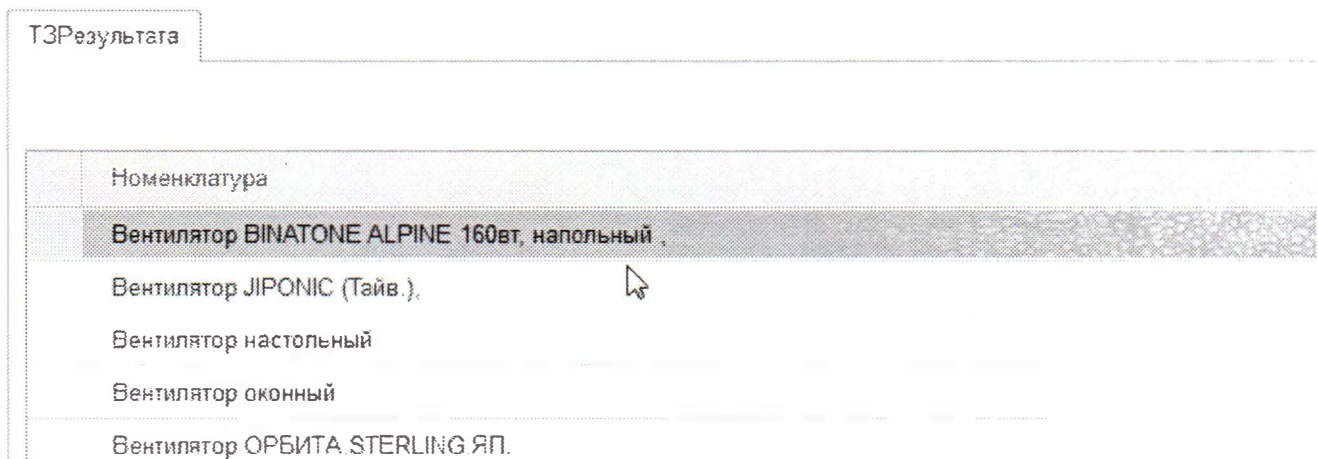


Рисунок 284

Теперь для задания шаблона используем специальный символ «_». Он означает один любой произвольный символ. Изменим наше условие на такое (Рисунок 285).

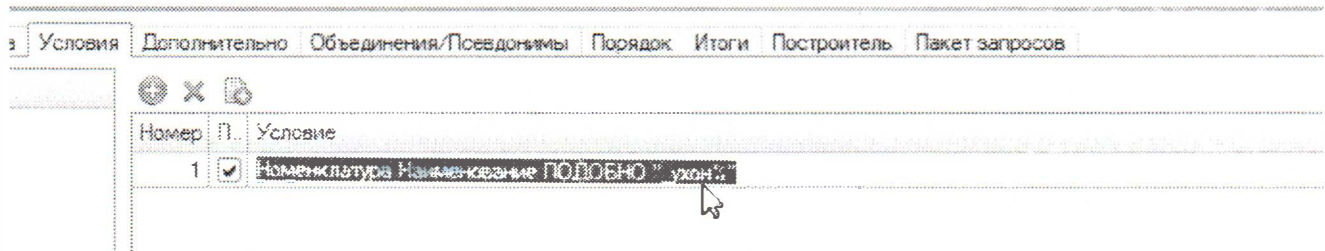


Рисунок 285

Мы получим всю номенклатуру, в наименовании которой присутствует подстрока «ухон», но при этом перед ней стоит какой-либо один символ. Так как справа от подстроки может быть любое количество любых символов, в результат попала только одна номенклатура, отвечающая данному условию (Рисунок 286).

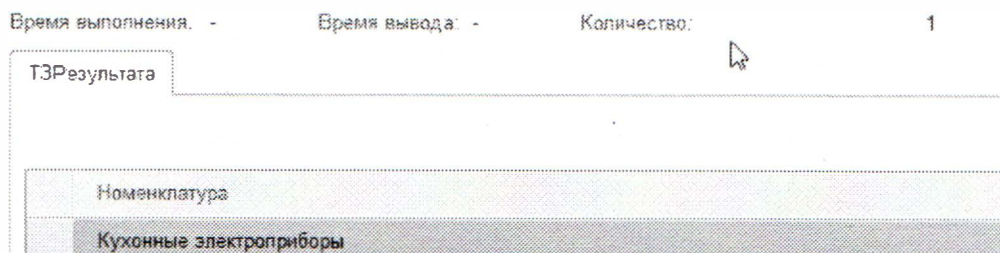


Рисунок 286

Давайте изменим условие, поставив два символа «__» в начале шаблона, и выполним запрос. Видим, что ни одна номенклатура не попала в результат. Но мы изменим наименование номенклатуры «Кухонные электроприборы», дописав в начале цифру 1 (Рисунок 287).

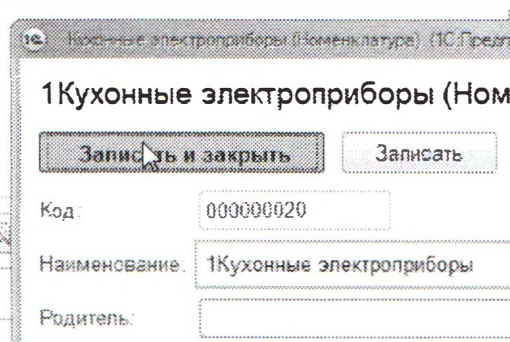


Рисунок 287

Снова выполним запрос.

Видим, что номенклатура с измененным названием попала в таблицу. Все правильно, подстроке «ухон» предшествуют 2 символа.

Используем специальные символы «[]», которые используются для задания одиночных символов, списка символов или диапазона символов. Зададим такое условие (Рисунок 288).

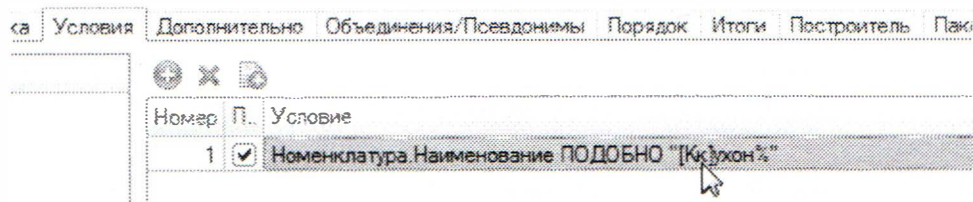


Рисунок 288

Это значит, что впереди может быть один любой символ из перечисленных в квадратных скобках. То есть либо «К», либо «к». Но обязательно один.

Выполнив запрос с таким условием, получим (Рисунок 289).



Рисунок 289

Теперь зададим в квадратных скобках диапазон. Это значит, что должен быть один символ, но уже из диапазона. Диапазон задается начальным символом дефисом и конечным символом диапазона. Концы диапазона включены в него.

Мы зададим диапазон символов от «а» до «я», то есть все буквы алфавита (Рисунок 290).

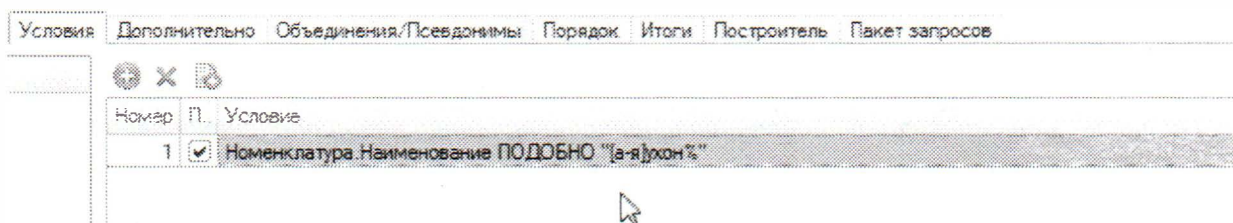


Рисунок 290

Выполним запрос и убедимся, что номенклатура «Кухонные приборы» опять попадает в результат. Если же мы зададим диапазон от «а» до «б», то уже ни одна номенклатура не попадет в результат. Если же мы изменим наименование этой номенклатуры таким образом, то она попадет в результат, что подтверждает, что концы диапазона также включены в него (Рисунок 291).

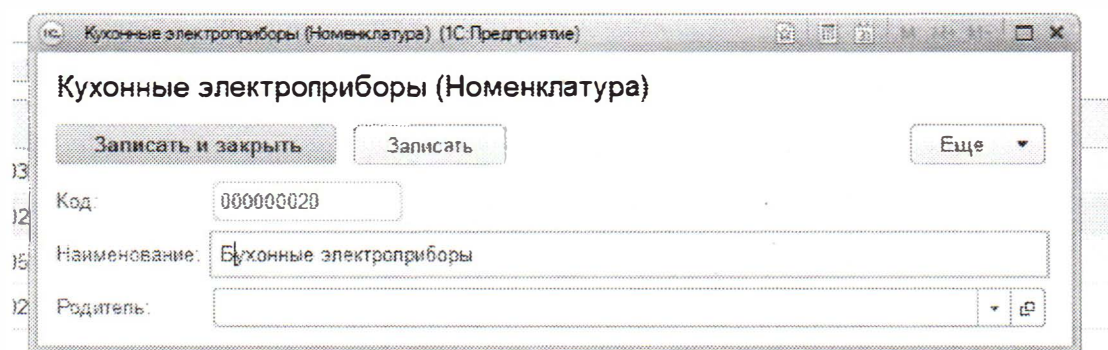


Рисунок 291

3.13 Функции языка. Условия. И, ИЛИ, Не.

Последними в этой главе рассмотрим логические операторы «И», «ИЛИ» и «НЕ». Выберем номенклатуру и количество из регистра накопления «Товары на складах». Зададим условие на количество: оно должно быть равно 150 или 27 (Рисунок 292).

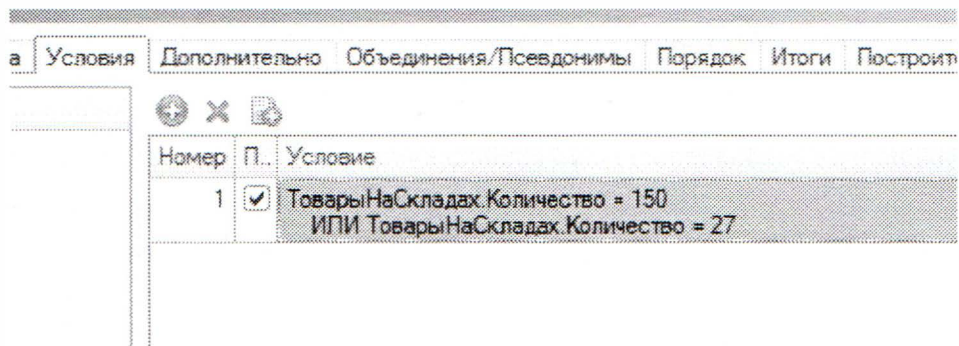


Рисунок 292

То есть, если выполняется хотя бы одно из этих условий, то строка попадает в результат (Рисунок 293).

Количество	Номенклатура
150	Вентилятор настольный
27	Кондиционер ELEKTA
150	Вентилятор ОРБИТА, STERLING, ЯП

Рисунок 293

Выполнив запрос, мы в этом убеждаемся.

Оператор не инвертирует (делает противоположным) условие. То есть если условие выполнилось (истина), то оно станет ложью (не выполнится).

Сделаем это с нашим последним условием. То есть в выборку должны попасть все строки с количеством не равным 150 или с количеством не равным 27.

При этом прежнее условие мы возьмем в скобки, которые определяют приоритет выполнения операций, как в обычной математике. Сначала выполнится проверка на равенство 150 или 27, а затем оператор «НЕ» или инверсия (отрицание) (Рисунок 294).

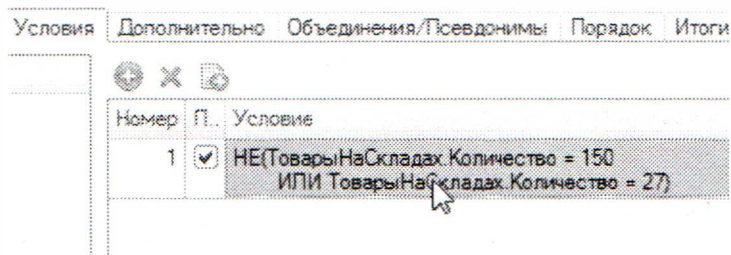


Рисунок 294

Выполним запрос и убедимся, что среди строк нет ни одной с количеством 150 или 27 (Рисунок 295).

Количество	Номенклатура
15	Вентилятор оконный
20	Кондиционер ELEKTRON
34	Вентилятор BRATONE ALPINE 150Вт, напольный
32	Вентилятор JIPONIC (Таив.)
40	Вентилятор настольный

Рисунок 295

Логический оператор «И» означает, что должны быть выполнены оба условия. В дальнейшем все это мы будем отрабатывать в практической части и домашних заданиях. Кроме того, описание всех функций можно посмотреть в теоретической части к курсу.

Глава 4.

4.1 Параметры виртуальных таблиц на примере условия запроса. Временные таблицы.

Рассмотрим использование параметров виртуальных таблиц на примере условий запроса. Предположим, стоит задача получить данные о приходах и расходах по всем товарам, на которые оформлялись заказы покупателей. Прежде всего, требуется зайти в режим «Предприятие,8» учебной информационной базы (https://yadi.sk/d/fhh_Vhcr8ueCNg) и открыть консоль запросов (Рисунок 296):

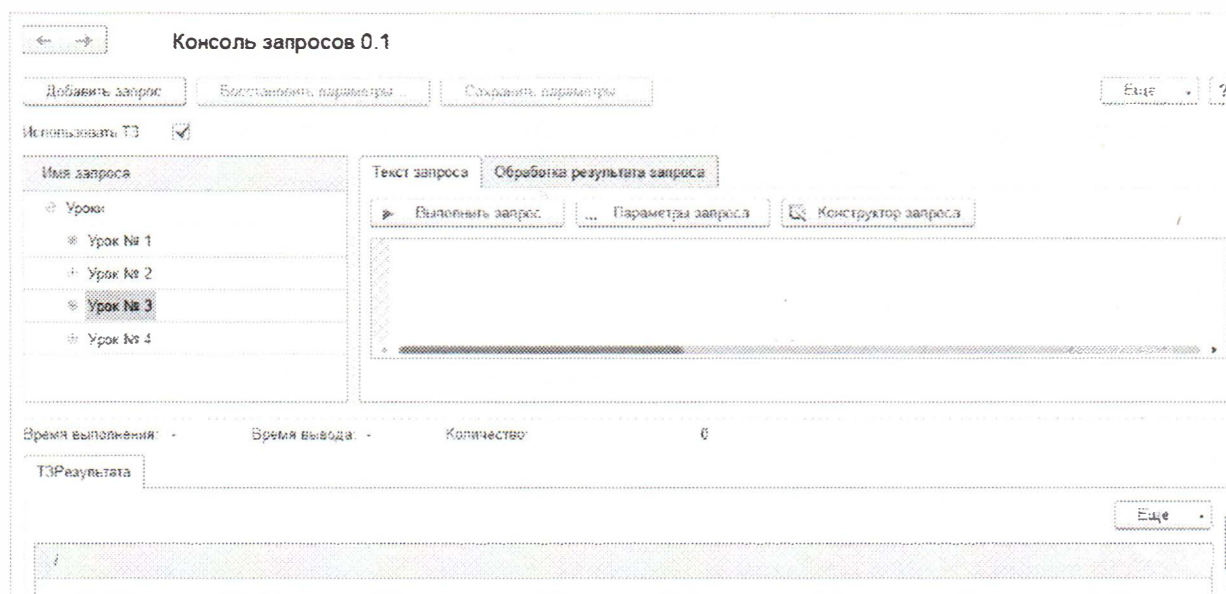


Рисунок 296

Задачу выполним в 2 этапа. В первом – создадим вспомогательный запрос. Источником данных для вспомогательного запроса послужит виртуальная таблица оборотов регистра накопления «Заказы покупателей», из которой мы получим все данные по полю «Номенклатура» (Рисунок 297):

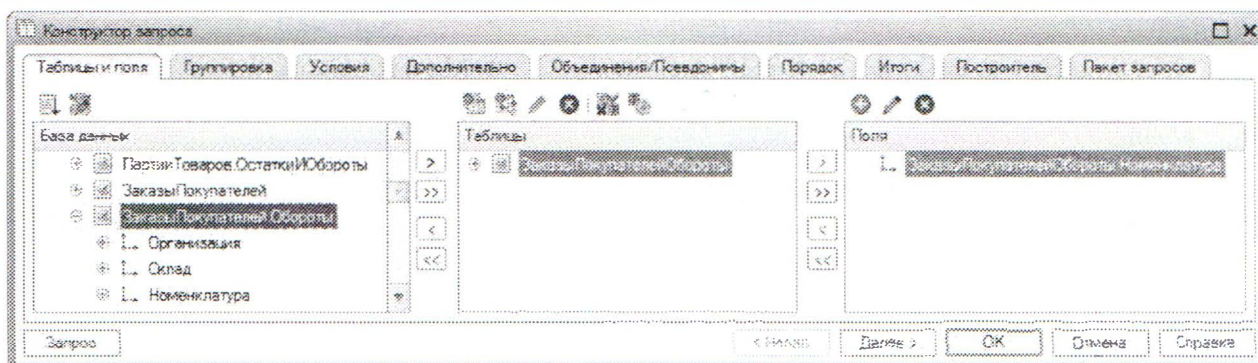


Рисунок 197

Текст вспомогательного запроса:

ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателейОбороты

Результат выполнения вспомогательного запроса (Рисунок 298):

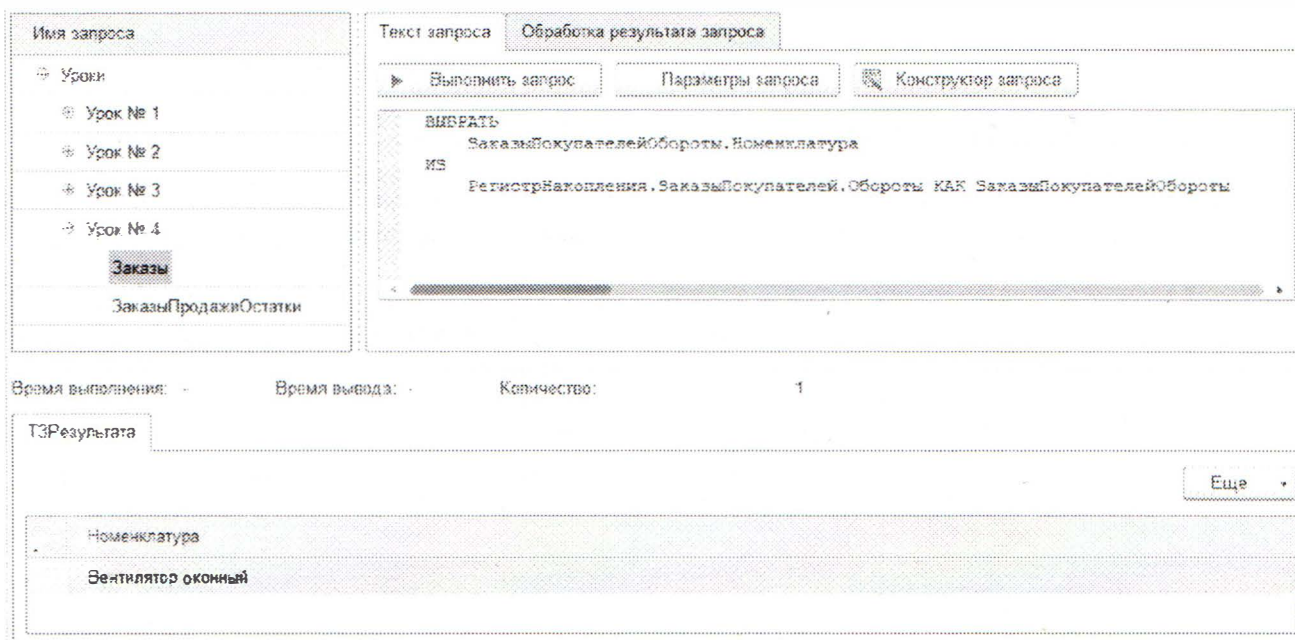


Рисунок 298

Следующим этапом получим данные о приходах и расходах по складу по всем товарам, на которые оформлялись заказы покупателей. Для этого обратимся к источникам данных (Рисунок 299):

- Таблица остатков регистра «Заказы покупателей»;
- Таблица оборотов регистра «Продажи»;
- Таблица оборотов регистра «Товары на складах».

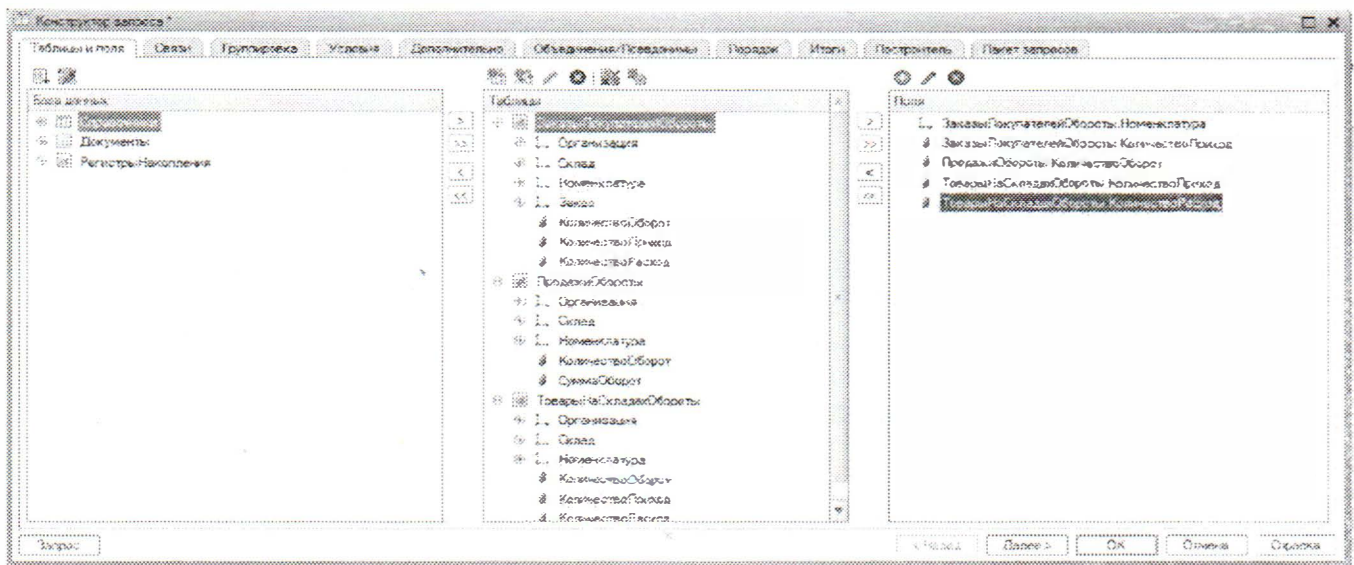


Рисунок 299

Здесь нас интересуют данные, отображенные в таблице 7.

Таблица 7 – Источник данных, поля выборки, комментарий

Источник данных	Поля выборки	Комментарий
Заказы покупателей. Обороты.	Номенклатура, Количество, Приход.	Для получения данных о номенклатуре по заказам покупателей.
Продажи. Обороты.	Количество. Оборот.	Для получения данных о проданной номенклатуре
Товары на складах. Обороты.	Количество приход. Количество расход.	Для получения данных об остатках номенклатуры на складах

На закладке «Связи» связываем таблицы левым соединением по полю «Номенклатура» (Рисунок 300):

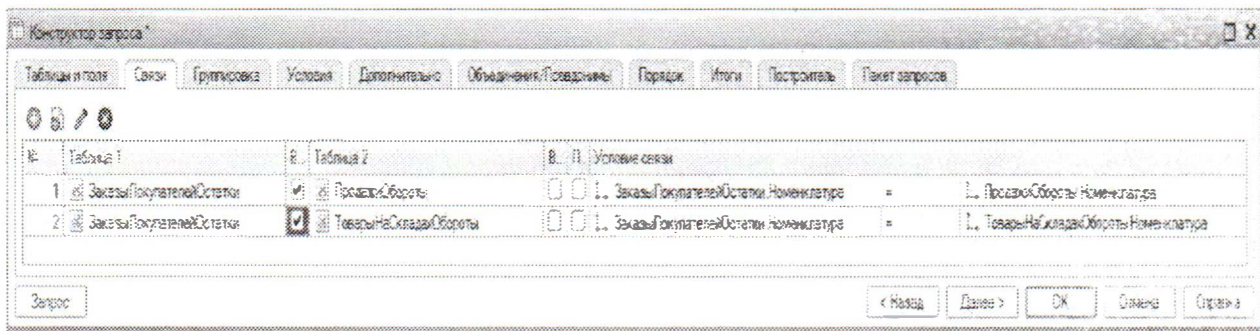


Рисунок 300

Для обработки отсутствующего значения в результате соединения таблиц (не найдено значения в одной из таблиц при связи по полям соединения), рекомендуется использовать

проверку на NULL. Для этого применяется функция языка запросов «ЕСТЬNULL» (Рисунок 301).

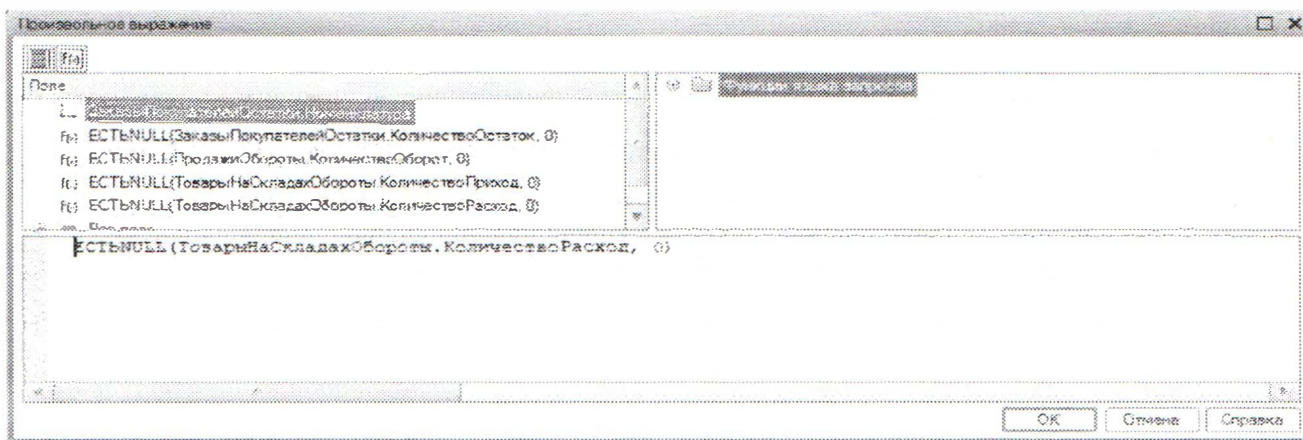


Рисунок 301

Для удобства работы с результатом запроса, рекомендуется задать псевдонимы полей на вкладке «Объединения/Псевдонимы» (Рисунок 302):

- «Заказано» – количество номенклатуры по заказам покупателей;
- «Продано» – количество проданной номенклатуры;
- «Пришло на склад» – количество поступившей на склад номенклатуры;
- «Ушло со склада» – количество реализованной номенклатуры со склада.

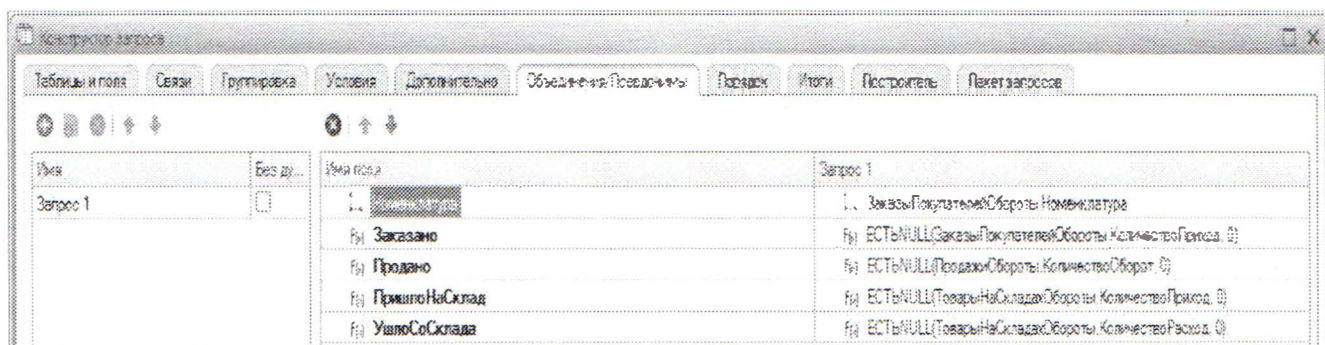


Рисунок 302

Текст запроса будет выглядеть следующим образом:

ВЫБРАТЬ

ЗаказыПокупателей.Обороты.Номенклатура,

«ЕСТЬNULL»(ЗаказыПокупателей.Обороты.КоличествоПриход, 0) КАК Заказано,

«ЕСТЬNULL»(Продажи.Обороты.КоличествоОборот, 0) КАК Продано,

«ЕСТЬNULL»(ТоварыНаСкладах.Обороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

«ЕСТЬNULL»(ТоварыНаСкладах.Обороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты

Проанализируем текст с точки зрения оптимизации. Как видим, данная конструкция не очень оптимальна, так как¹ при исполнении запроса сначала формируются данные из таблиц-источников по всей номенклатуре, без задания параметров, и только потом из этого массива отбираются нужные, на основе связывания таблиц.

Рекомендуется выполнить настройки запроса таким образом, чтобы изначально лишние записи в выборку не попадали. Для этого выполнить следующие действия:

1. Задать параметры виртуальных таблиц;
2. Использовать конструкцию вложенных запросов.

Зададим параметры виртуальных таблиц. Для этого требуется выделить источник данных в конструкторе запроса и вызвать контекстное меню правой кнопкой мыши (Рисунок 303).

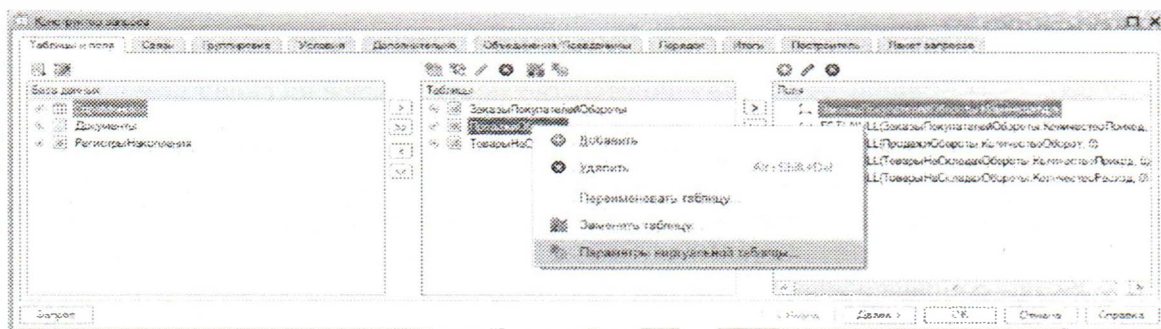


Рисунок 303

В форме параметров таблиц «ПродажиОбороты» и «ТоварыНаСкладахОбороты» необходимо заполнить данные по условию. В качестве условия используется вспомогательный запрос, созданный на первом этапе (Рисунок 304-305):

ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателейОбороты

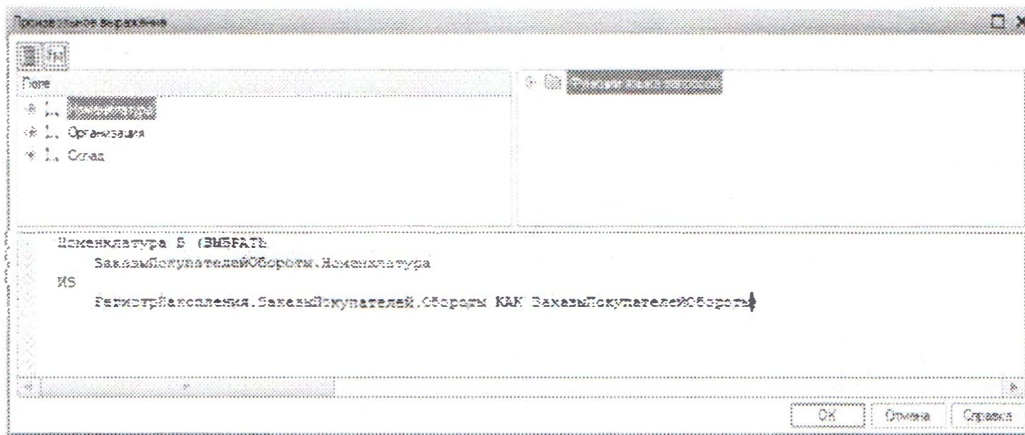


Рисунок 304

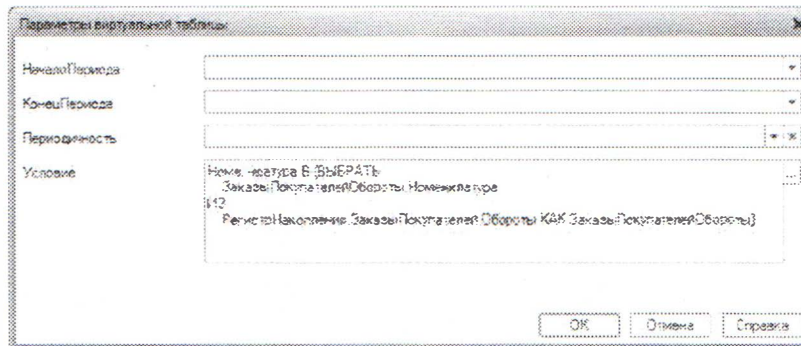


Рисунок 305

В результате задания параметров, в процессе исполнения запроса, данные по регистрам «Продажи» и «ТоварыНаСкладах» запрашиваются только по той номенклатуре, которая есть в заказах, что сокращает время исполнения запроса. Однако и такой вариант запроса не является оптимальным решением задачи, так как условие:

Номенклатура В

(ВЫБРАТЬ

ЗаказыПоккупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПоккупателей.Обороты КАК ЗаказыПоккупателейОбороты)

будет выполняться в каждой таблице, в которой оно задано. Иными словами, вспомогательный запрос выполнится трижды: в таблице «ЗаказыПоккупателейОбороты», таблице «ПродажиОбороты» и таблице «ТоварыНаСкладахОбороты». Не логичнее ли вынести исполнение данного кода в отдельный блок? Посмотрим, как решается задача с использованием вложенного запроса. Для этого нужно создать новую таблицу и скопировать текст условия (Рисунок 306).

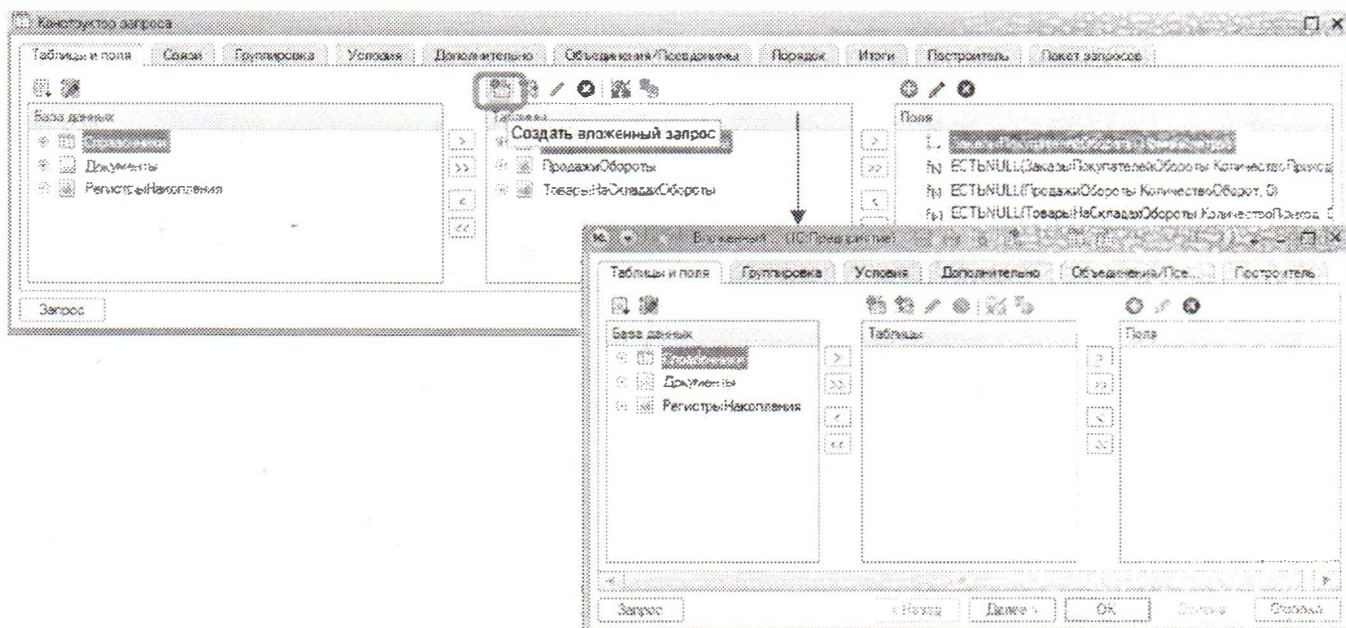


Рисунок 306

В поле текста запроса вставляем фрагмент условия (Рисунок 307).

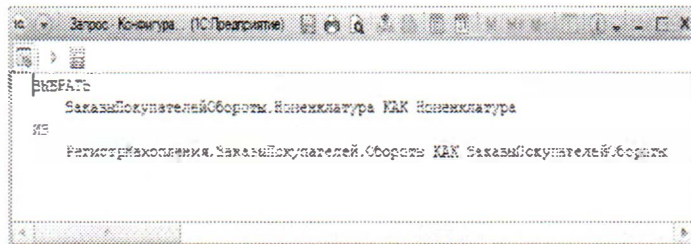


Рисунок 307

4.2 Менеджер Временных Таблиц.

Сложность практического применения вложенного запроса в том, что нельзя его использовать в дальнейшем, как условие, и потребуются прописывать еще одно соединение таблиц. В связи с этим, предлагается использовать такой объект встроенного языка, как «Менеджер Временных Таблиц», который позволяет передавать временные таблицы из одного запроса в другой. Рассмотрим на примере.

Для этого потребуется открыть режим «Конфигуратор», создать новый отчет (назовем его «Временные таблицы») и перейти в модуль формы отчета (Рисунок 308):

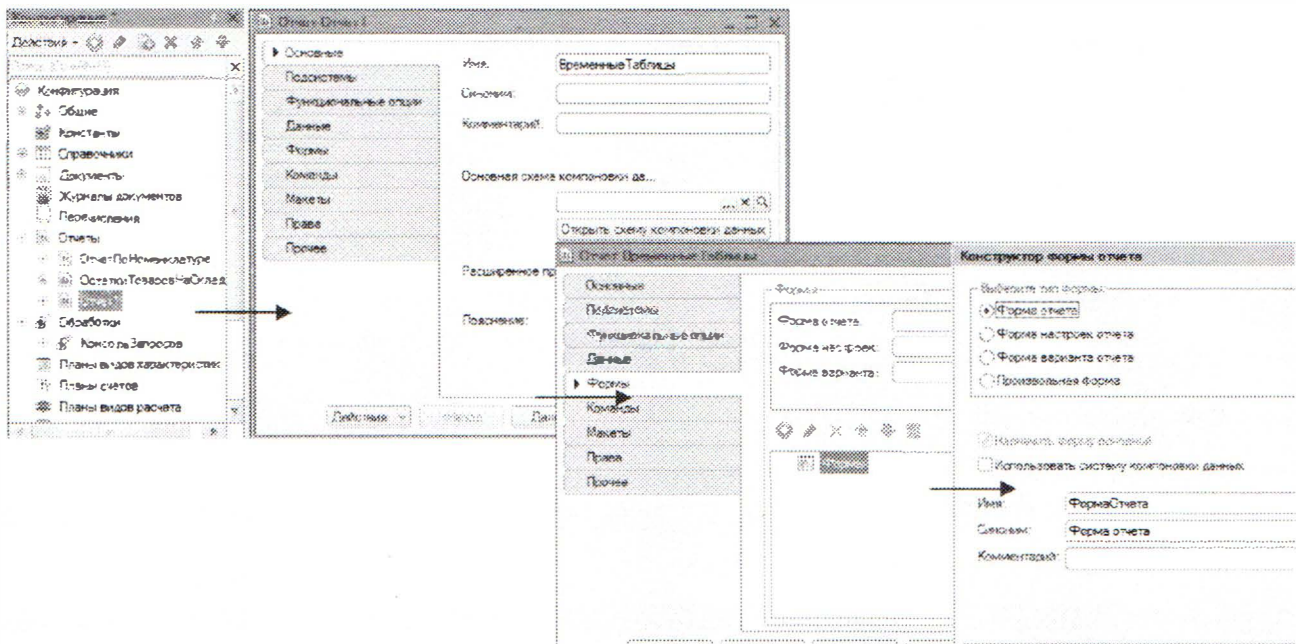


Рисунок 308

В форме отчета нужно создать реквизит с типом данных «ТабличныйДокумент» и команду «Сформировать», в параметрах которой укажем ссылку на реквизит (Рисунок 309-310):

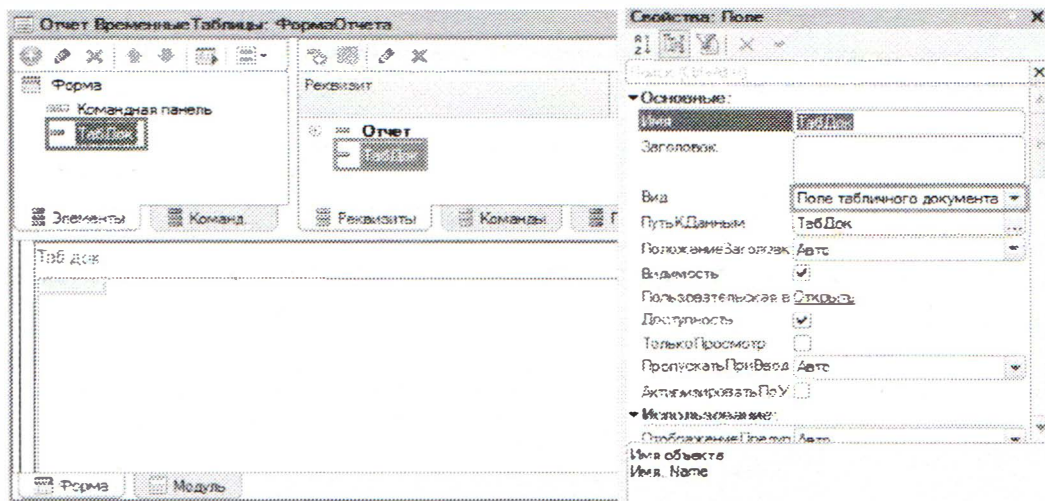


Рисунок 309

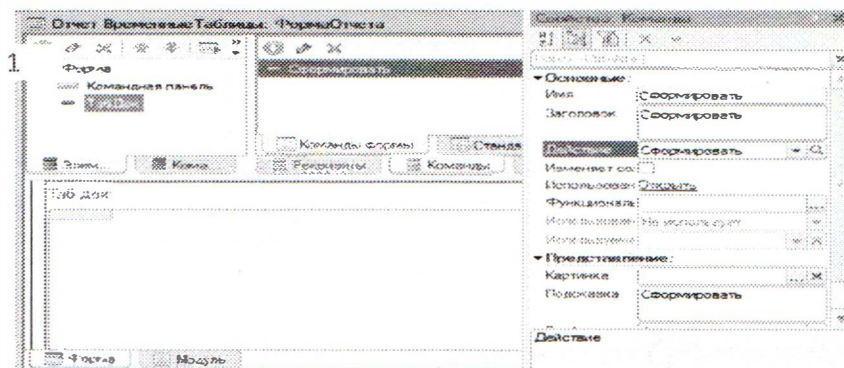


Рисунок 310

В процедуре «СформироватьНаСервере(ТабДок)» вставим следующий код:

&НаКлиенте

Процедура Сформировать(Команда)

СформироватьНаСервере(ТабДок);

КонецПроцедуры

&НаСервере

Процедура СформироватьНаСервере(ТабДок)

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.Текст = "";

КонецПроцедуры

Далее воспользуемся конструктором запроса, в текстовое поле которого скопируем код вложенного запроса (Рисунок 311):

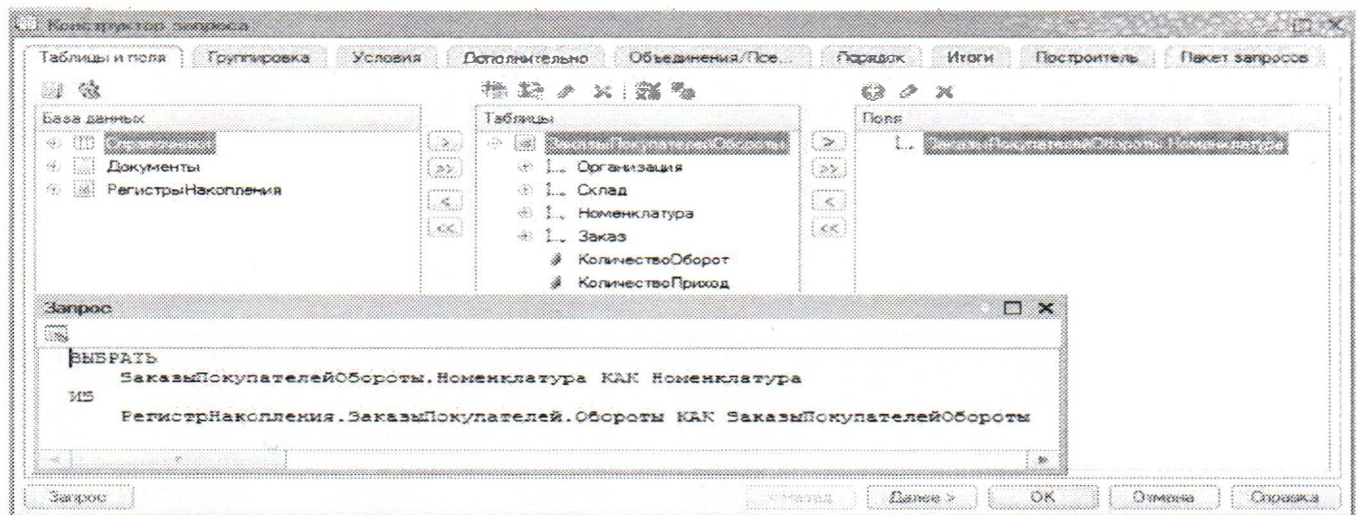


Рисунок 311

Перейдем на закладку «Дополнительно» конструктора запросов и укажем тип запроса – «Создание временной таблицы» (Рисунок 312):

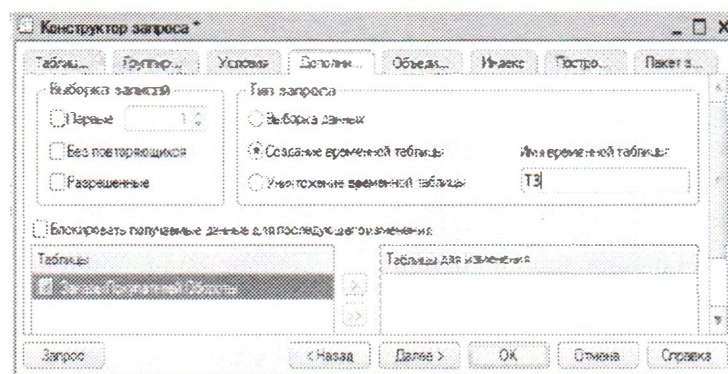


Рисунок 312

Код запроса выглядит следующим образом:

&НаСервере

Процедура СформироватьНаСервере(ТабДок)

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура КАК Номенклатура

|ПОМЕСТИТЬ ТЗ

|ИЗ

| РегистрНакопления.ЗаказыПокупателей.Обороты КАК

ЗаказыПокупателейОбороты";

Запрос.Выполнить());

КонецПроцедуры

После выполнения метода «Выполнить()» в «МенеджерВременныхТаблиц» будет помещена временная таблица «ТЗ», с которой и будем работать. Для примера допишем и проанализируем код (Рисунок 313):

1. Создадим текст запроса, в который скопируем ранее созданный запрос по продажам и остаткам, и перейдем к описанию временной таблицы:

Текст запроса:

ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура,

ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,

«ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты КАК

ЗаказыПокупателейОбороты)) КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты КАК

ЗаказыПокупателейОбороты)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ТоварыНаСкладахОбороты.Номенклатура

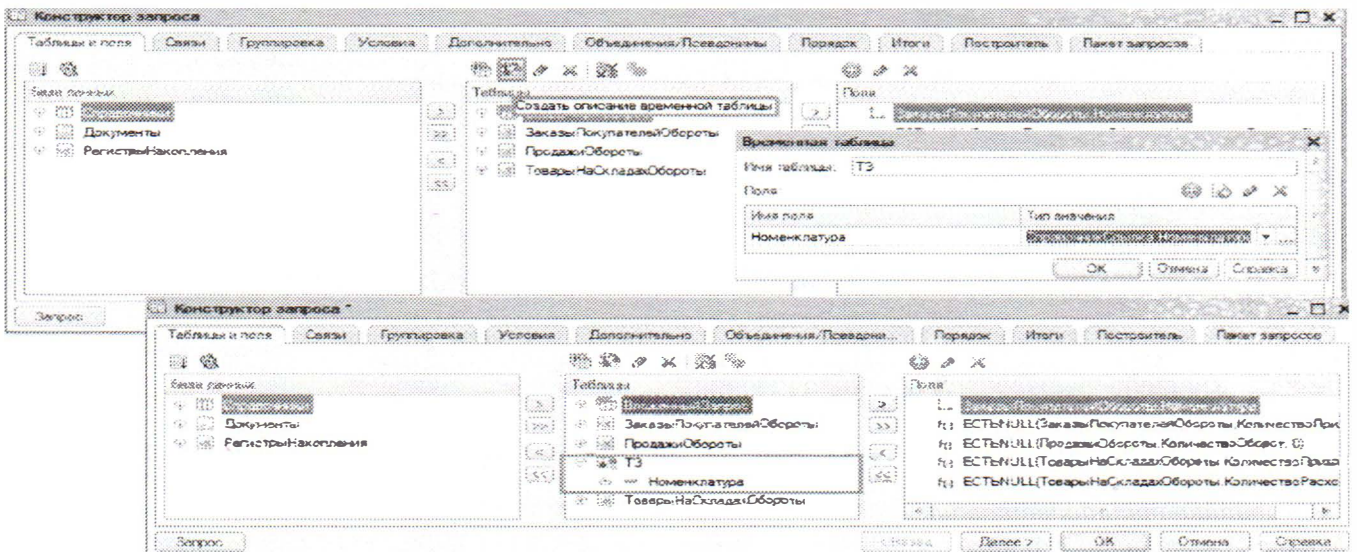


Рисунок 313

2. Отредактируем условие в параметрах таблиц (Рисунок 314).

Номенклатура В

(ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателейОбороты)

Заменить на:

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)

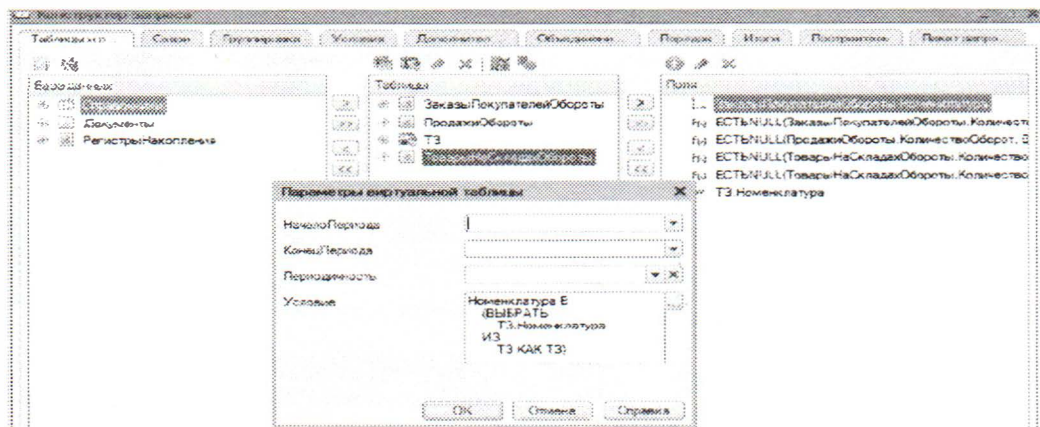


Рисунок 314

3. Поставим точку останова на методе «Запрос.Выполнить().Выгрузить()» и проанализируем «ТаблицуЗначений» (Рисунок 315).

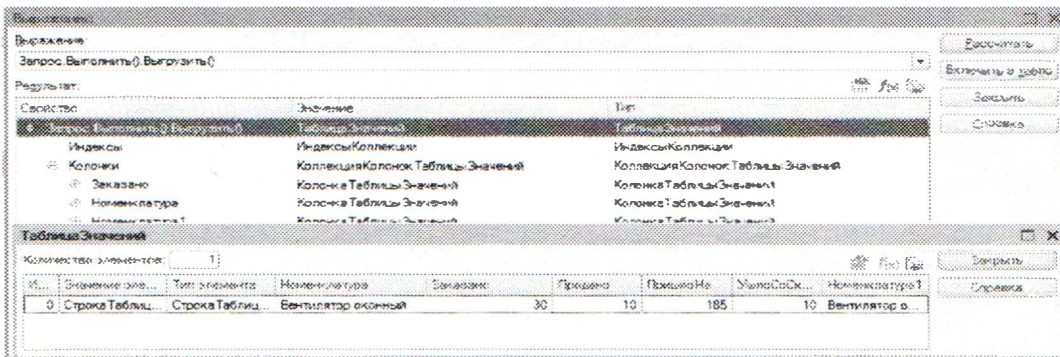


Рисунок 315

Таким образом, в процедуре «СформироватьНаСервере(ТабДок)» выполнен сначала текст запроса, переданный во временную таблицу, а при исполнении второго запроса в условии используется уже результат обработки временной таблицы. Ниже приведен код:

&НаСервере

Процедура СформироватьНаСервере(ТабДок)

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

| ЗаказыПокупателейОбороты.Номенклатура КАК Номенклатура

| ПОМЕСТИТЬ ТЗ

| ИЗ

| РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателей(Обороты);

Запрос.Выполнить();

Запрос.Текст = "ВЫБРАТЬ

| ЗаказыПокупателей(Обороты).Номенклатура,

| «ЕСТЬNULL»(ЗаказыПокупателей(Обороты).КоличествоПриход, 0) КАК Заказано,

| «ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

| «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

| «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада,

| ТЗ.Номенклатура КАК Номенклатура1

| ИЗ

| РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателей(Обороты

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

|

|

|

|

|

|

|

|

|

|

|

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура =
 ТоварыНаСкладахОбороты.Номенклатура,
 ТЗ КАК ТЗ";
 Запрос.Выполнить().Выгрузить();
 КонецПроцедуры

Усложним данный пример, для этого переименуем второй запрос (изменим наименование переменной с «Запрос» на «Запрос 2») для понимания принципа работы «МенеджераВременныхТаблиц» (Рисунок 316):

```

Планет ВременныхТаблиц.ФорматОтчета
Запрос1.Текст = "ИЗНАЧАТЬ
; ЗаказыПокупателейОбороты.Номенклатура КАК Номенклатура
; ДОБЕЖИТЬ ТЗ
; ИЗ
; РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателейОбороты";
Запрос1.Выполнить();
Запрос2 = Новый Запрос();
Запрос2.Текст = "ИЗНАЧАТЬ
; ЗаказыПокупателейОбороты.Номенклатура,
; КУПИТЬ1.ТоварыНаСкладахОбороты.КоличествоВКлад, С) КАК Заказано,
; КУПИТЬ2.ТоварыНаСкладахОбороты.КоличествоОборот, С) КАК Пролетено,
; КУПИТЬ3.ТоварыНаСкладахОбороты.КоличествоСклад, С) КАК ПролетеноСклад,
; ТЗ.Номенклатура КАК Номенклатура1
; ИЗ
; РегистрНакопления.ЗаказыПокупателей.Обороты(, , ) КАК ЗаказыПокупателейОбороты
; ЛЕВЫЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(
;
;
;
;
; Номенклатура В
; ИМБЕРАТЬ
; ТЗ.Номенклатура
; ИЗ
; ТЗ КАК ТЗ;) КАК ПролетеноОбороты
; ИС ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура
; ЛЕВЫЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(
;
;
;
;
; Номенклатура В
; ИМБЕРАТЬ
; ТЗ.Номенклатура
; ИЗ
; ТЗ КАК ТЗ;) КАК ТоварыНаСкладахОбороты
; ИС ЗаказыПокупателейОбороты.Номенклатура = ТоварыНаСкладахОбороты.Номенклатура
; ТЗ КАК ТЗ";
Запрос2.Выполнить().Выгрузить();
  
```

Рисунок 316

При исполнении будет выведена ошибка, смысл которой в отсутствии «МенеджераВременныхТаблиц» для «Запроса 2» (Рисунок 317):

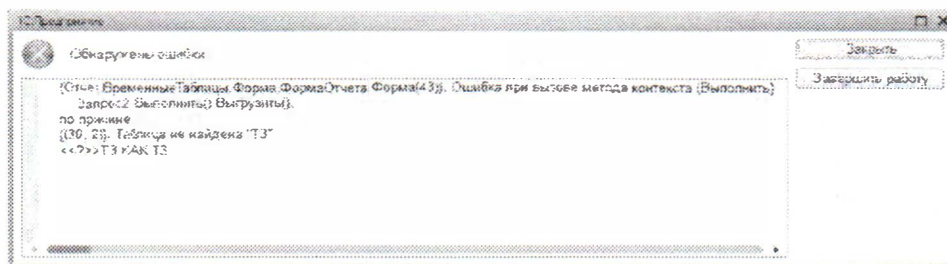


Рисунок 317

Для устранения ошибки потребуется изменить код процедуры в части работы с «МенеджеромВременныхТаблиц», и выполнить это можно несколькими вариантами.

Рассмотрим их.

Вариант 1. Присвоить «МенеджеруВременныхТаблиц» значение ранее созданной переменной.

Зададим «МенеджерВременныхТаблиц» в какой-либо переменной и укажем его в Запросе 2 (Рисунок 318):

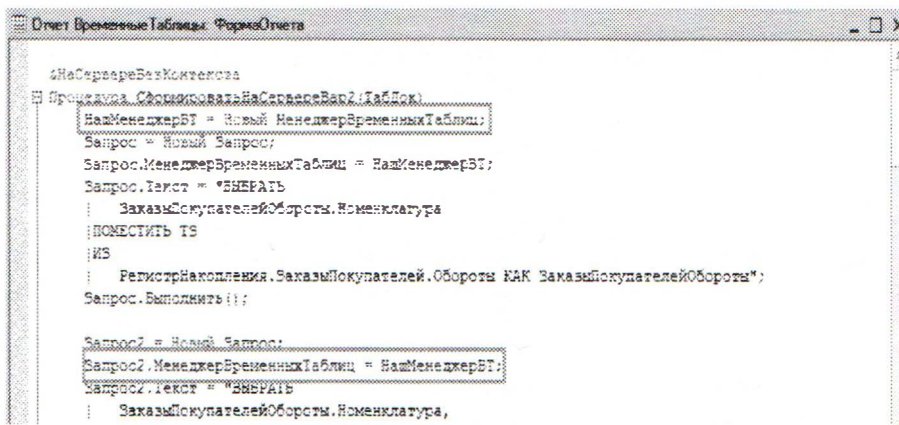


Рисунок 318

Таким образом, в первом запросе создаем объект «МенеджерВременныхТаблиц», а во втором запросе присваиваем это значение. Запрос выполняется корректно.

Вариант 2. Передать «МенеджеруВременныхТаблиц» «ТаблицуЗначений».

2.1. Для этого используем уже написанную ранее временную таблицу «ТЗ», но отредактируем ее следующим образом:

1. Используем временную таблицу «ТЗ» как параметр, добавив символ «&» и определив значение параметра методом «УстановитьПараметр()» (Рисунок 319).

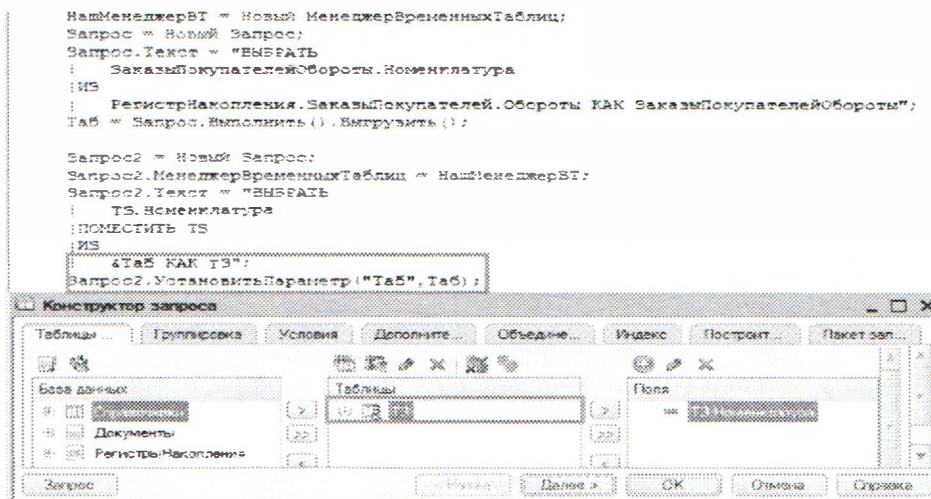


Рисунок 319

Передавать параметр будем из запроса «Таб», для этого отредактируем его, удалив «МенеджерВременныхТаблиц» и поменяв тип запроса «Выборка данных» (Рисунок 320):

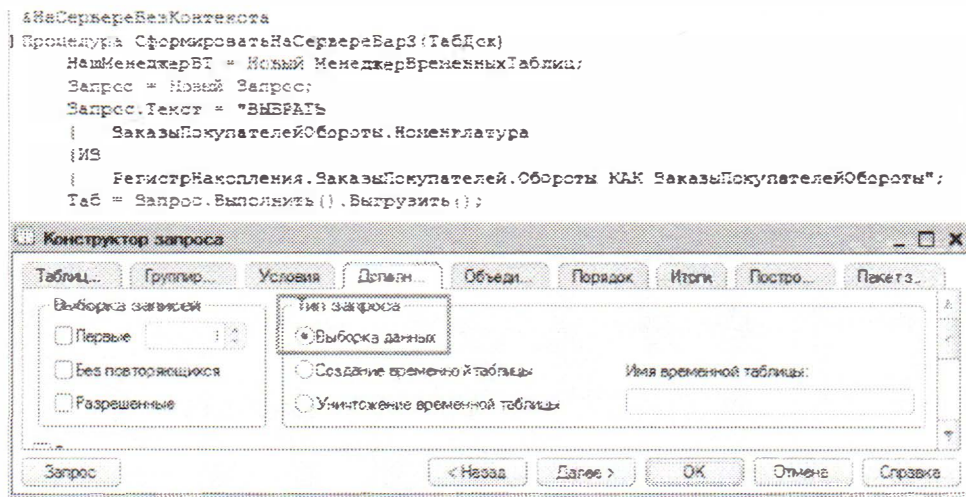


Рисунок 320

В результате выполнения запроса получим «ТаблицуЗначений», которую и будем передавать в параметр. Но для этого нужно изменить запрос 2, создав в нем временную таблицу «ТЗ» (Рисунок 321):

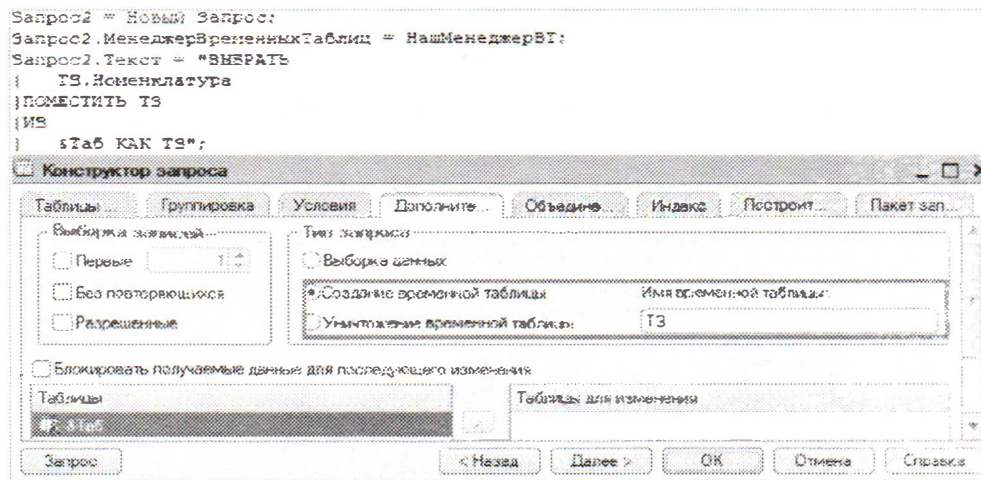


Рисунок 321

Текст запроса:

```

НашМенеджерВТ = Новый МенеджерВременныхТаблиц;
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|     ЗаказыПокупателейОбороты.Номенклатура
| ИЗ
|     РегистрНакопления.ЗаказыПокупателей.Обороты КАК ЗаказыПокупателейОбороты";
Таб = Запрос.Выполнить().Выгрузить();

```

```

Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = НашМенеджерВТ;
Запрос2.Текст = "ВЫБРАТЬ
|     ТЗ.Номенклатура
| ПОМЕСТИТЬ ТЗ
| ИЗ
|     &Таб КАК ТЗ";
Запрос2.УстановитьПараметр("Таб", Таб);
Запрос2.Выполнить();
Запрос2.Текст = "ВЫБРАТЬ

```

ЗаказыПокупателейОбороты.Номенклатура,
 ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,
 «ЕСТЬНИЛ»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,
 «ЕСТЬНИЛ»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,
 «ЕСТЬНИЛ»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСoSклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты
 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура =

ТоварыНаСкладахОбороты.Номенклатура,

ТЗ КАК ТЗ";

Запрос2.Выполнить().Выгрузить());

2.2. Теперь создадим таблицу значений не путем редактирования временной таблицы, а самостоятельно, с самого начала. Для этого в процедуре «СформироватьНаСервере(ТабДок)» выполним действия:

1. Создадим новый объект «ТаблицаЗначений» с колонкой «Номенклатура» и добавим в нее строку, которую получали ранее из запроса:

Таб = Новый ТаблицаЗначений;

Таб.Колонки.Добавить("Номенклатура");

НовСтр = Таб.Добавить();

НовСтр.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("000000049");

Поставим точку останова на строке «Запрос2.Выполнить()» и запустим приложение под отладкой (Рисунок 322).


```

&НаСервереБезКонтекста
Процедура СформироватьНаСервере(ТабДок)
//Таб = Новый ТаблицаЗначений;
//Таб.Колонки.Добавить("Номенклатура");
//НовСтр = Таб.Добавить();
//НовСтр.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("000000049");

Таб = Новый ТаблицаЗначений;
Массив = Новый Массив;
Массив.Добавить(Тип("СправочникСсылка.Номенклатура"));
ОписаниеТипов = Новый ОписаниеТипов(Массив);
Таб.Колонки.Добавить("Номенклатура", ОписаниеТипов);
НовСтр = Таб.Добавить();
НовСтр.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("000000049");

НашМенеджерВТ = Новый МенеджерВременныхТаблиц;
Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = НашМенеджерВТ;
Запрос2.Текст = "ВЫСПАТЬ
| ТЗ.Номенклатура
| ПОМЕСТИТЬ ТЗ
| ИЗ
| &Таб КАК ТЗ";
Запрос2.УстановитьПараметр("Таб", Таб);
Запрос2.Выполнить();
Запрос2.Текст = "ВЫБРАТЬ
| ЗаказыПокупателейОборота.Номенклатура,
| ЗаказыПокупателейОборота.КоличествоПриход КАК Заказано,
| ЕСТЬNULL(ПродажиОборота.КоличествоОборот, 0) КАК Продано,
| ЕСТЬNULL(ТоварыНаСкладахОборота.КоличествоПриход, 0) КАК ПришлоНаСклад,
| ЕСТЬNULL(ТоварыНаСкладахОборота.КоличествоРасход, 0) КАК УшлоСоСклада
| ИЗ

```

Рисунок 324

Запустим приложение под отладчиком и проанализируем результат (Рисунок 325):

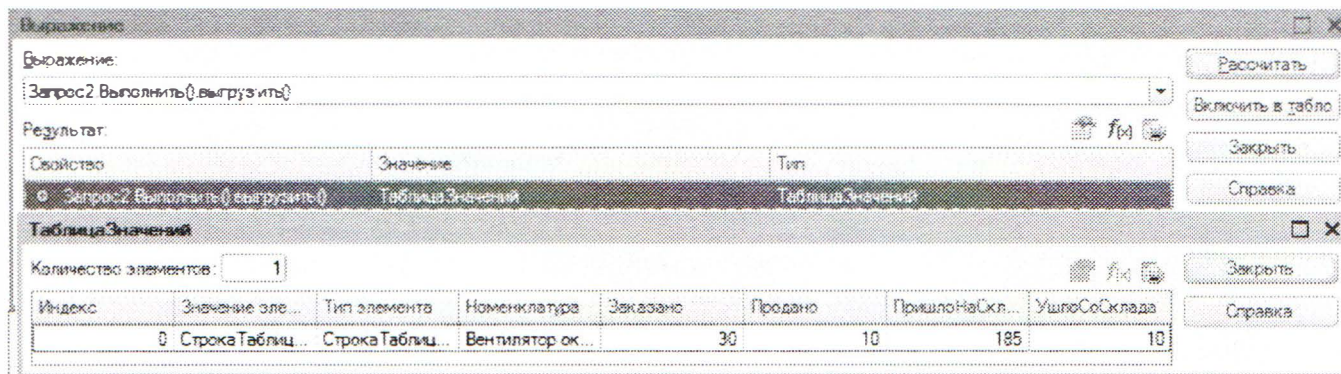


Рисунок 325

Запрос выполнен корректно.

Текст процедуры:

```

&НаСервереБезКонтекста
Процедура СформироватьНаСервере(ТабДок)
Таб = Новый ТаблицаЗначений;
Массив = Новый Массив;
Массив.Добавить(Тип("СправочникСсылка.Номенклатура"));
ОписаниеТипов = Новый ОписаниеТипов(Массив);
Таб.Колонки.Добавить("Номенклатура", ОписаниеТипов);
НовСтр = Таб.Добавить();
НовСтр.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("000000049");

```

НашМенеджерВТ = Новый МенеджерВременныхТаблиц;

Запрос2 = Новый Запрос;

Запрос2.МенеджерВременныхТаблиц = НашМенеджерВТ;

Запрос2.Текст = "ВЫБРАТЬ

 ТЗ.Номенклатура

ПОМЕСТИТЬ ТЗ

ИЗ

 &Таб КАК ТЗ";

Запрос2.УстановитьПараметр("Таб",Таб);

Запрос2.Выполнить();

Запрос2.Текст = "ВЫБРАТЬ

 ЗаказыПокупателейОбороты.Номенклатура,

 ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,

 «ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

 «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

 «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

 РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

 ,

 ,

 ,

 Номенклатура В

 (ВЫБРАТЬ

 ТЗ.Номенклатура

 ИЗ

 ТЗ КАК ТЗ)) КАК ПродажиОбороты

 ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

 ,

 ,

 ,

 Номенклатура В

 (ВЫБРАТЬ

 ТЗ.Номенклатура

 ИЗ

 ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты

 ПО ЗаказыПокупателейОбороты.Номенклатура =

ТоварыНаСкладахОбороты.Номенклатура,

 ТЗ КАК ТЗ";

Запрос2.Выполнить();

Запрос2.Текст = "ВЫБРАТЬ

 ЗаказыПокупателейОбороты.Номенклатура,

 ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,

 «ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

 «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

 «ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

 РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

 ,

 ,

 ,

 Номенклатура В

 (ВЫБРАТЬ

 ТЗ.Номенклатура

 ИЗ

ТЗ КАК ТЗ)) КАК ПродажиОбороты
 ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура
 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

Номенклатура В
 (ВЫБРАТЬ
 ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура =

ТоварыНаСкладахОбороты.Номенклатура,

ТЗ КАК ТЗ";

Запрос2.Выполнить());

4.3 Уничтожение временных таблиц.

Рассмотрим ситуацию необходимости уничтожения временных таблиц.

В случае, когда временная таблица используется однократно и занимает большой объем памяти, её целесообразно уничтожить после использования. Для этого в конструкторе запроса на закладке «Дополнительно» есть опция «Уничтожение временной таблицы» (Рисунок 326).

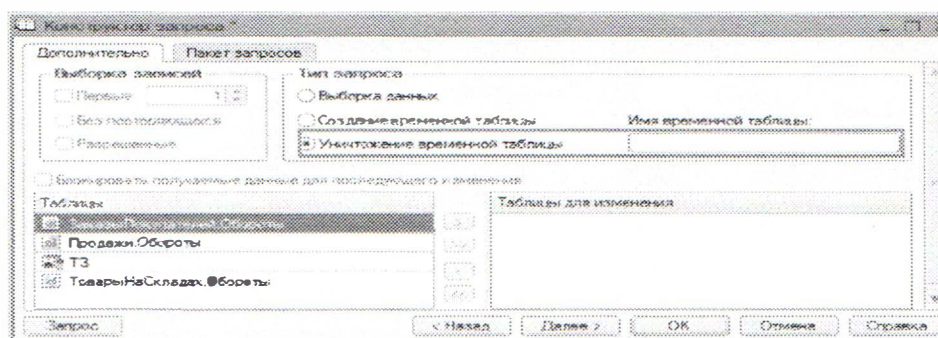


Рисунок 326

Ниже представлен текст процедуры:

&НаСервереБезКонтекста

Процедура СформироватьНаСервере(ТабДок)

Таб = Новый ТаблицаЗначений;

Массив = Новый Массив;

Массив.Добавить(Тип("СправочникСсылка.Номенклатура"));

ОписаниеТипов = Новый ОписаниеТипов(Массив);

Таб.Колонки.Добавить("Номенклатура",ОписаниеТипов);

НовСтр = Таб.Добавить();

НовСтр.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("000000049");

НашМенеджерВТ = Новый МенеджерВременныхТаблиц;

Запрос2 = Новый Запрос;

Запрос2.МенеджерВременныхТаблиц = НашМенеджерВТ;

Запрос2.Текст = "ВЫБРАТЬ

ТЗ.Номенклатура

ПОМЕСТИТЬ ТЗ

ИЗ

& Таб КАК ТЗ";

Запрос2.УстановитьПараметр("Таб", Таб);

Запрос2.Выполнить();

Запрос2.Текст = "ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура,

ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,

«ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура =

ТоварыНаСкладахОбороты.Номенклатура,

ТЗ КАК ТЗ";

Запрос2.Выполнить();

Запрос2.Текст = "УНИЧТОЖИТЬ ТЗ";

Запрос2.Выполнить();

Запрос2.Текст = "ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура,

ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,

«ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,

«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСоСклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

*ТЗ КАК ТЗ)) КАК ПродажиОбороты
ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(*

*Номенклатура В
(ВЫБРАТЬ
ТЗ.Номенклатура
ИЗ*

*ТЗ КАК ТЗ)) КАК ТоварыНаСкладахОбороты
ПО ЗаказыПокупателейОбороты.Номенклатура =
ТоварыНаСкладахОбороты.Номенклатура,
ТЗ КАК ТЗ";
Запрос2.Выполнить());
КонецПроцедуры*

Если после уничтожения временной таблицы мы попытаемся к ней обратиться, то будет выдана ошибка. Для того чтобы отследить результат выполнения процедуры, запустим приложение в режиме «1С:Предприятие» и сформируем отчет (Рисунок 327):

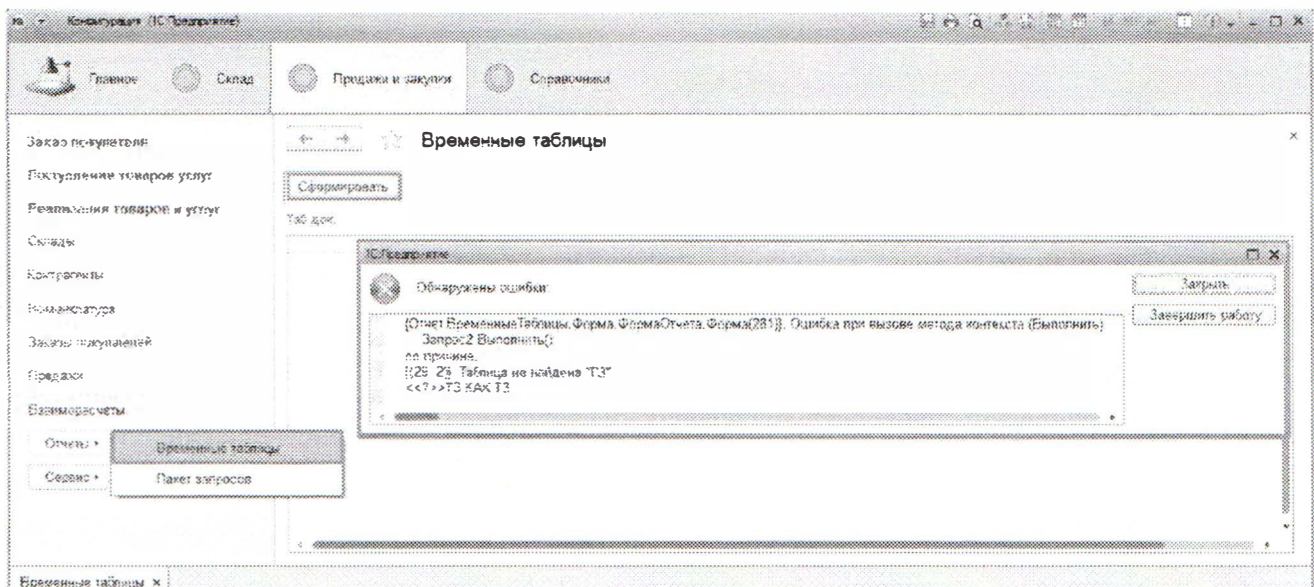


Рисунок 27

Можем убедиться, что при нажатии «Сформировать» будет выведена ошибка, так как обращение к временной таблице после ее уничтожения не является корректным.

4.4 Пакет запросов.

Рассмотрим решение данной задачи с использованием конструкции пакетных запросов. Для этого вернемся в консоль и создадим новый запрос «ПакетЗапросов», скопировав текст запроса из предыдущей главы «ЗаказыПродажиОстатки»:

ВЫБРАТЬ

*ЗаказыПокупателейОбороты.Номенклатура,
ЗаказыПокупателейОбороты.КоличествоПриход КАК Заказано,*

«ЕСТЬNULL»(ПродажиОбороты.КоличествоОборот, 0) КАК Продано,
«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоПриход, 0) КАК ПришлоНаСклад,
«ЕСТЬNULL»(ТоварыНаСкладахОбороты.КоличествоРасход, 0) КАК УшлоСoСклада

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты(, ,) КАК ЗаказыПокупателейОбороты
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты КАК

ЗаказыПокупателейОбороты)) КАК ПродажиОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ПродажиОбороты.Номенклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Обороты(

,

,

,

Номенклатура В

(ВЫБРАТЬ

ЗаказыПокупателейОбороты.Номенклатура

ИЗ

РегистрНакопления.ЗаказыПокупателей.Обороты КАК

ЗаказыПокупателейОбороты)) КАК ТоварыНаСкладахОбороты

ПО ЗаказыПокупателейОбороты.Номенклатура = ТоварыНаСкладахОбороты.Номенклатура

На вкладке «Пакет запросов» консоли отображился «Запрос пакета 1», который по умолчанию является первым запросом (Рисунок 328).

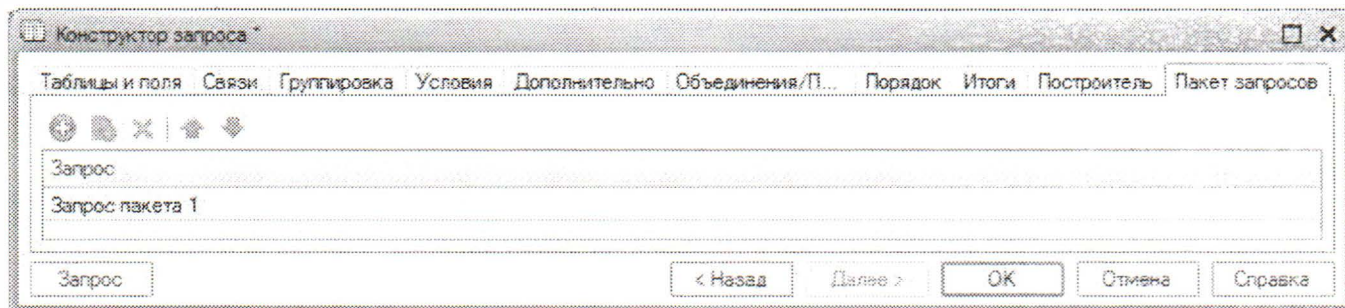


Рисунок 328

Создадим второй пакет запросов с помощью кнопки «Добавить», после чего на закладке «Таблицы и поля» добавится дополнительная вкладка запроса. Выберем поле «Номенклатура» из виртуальной таблицы оборотов регистра «Заказы покупателей» (Рисунок 329).

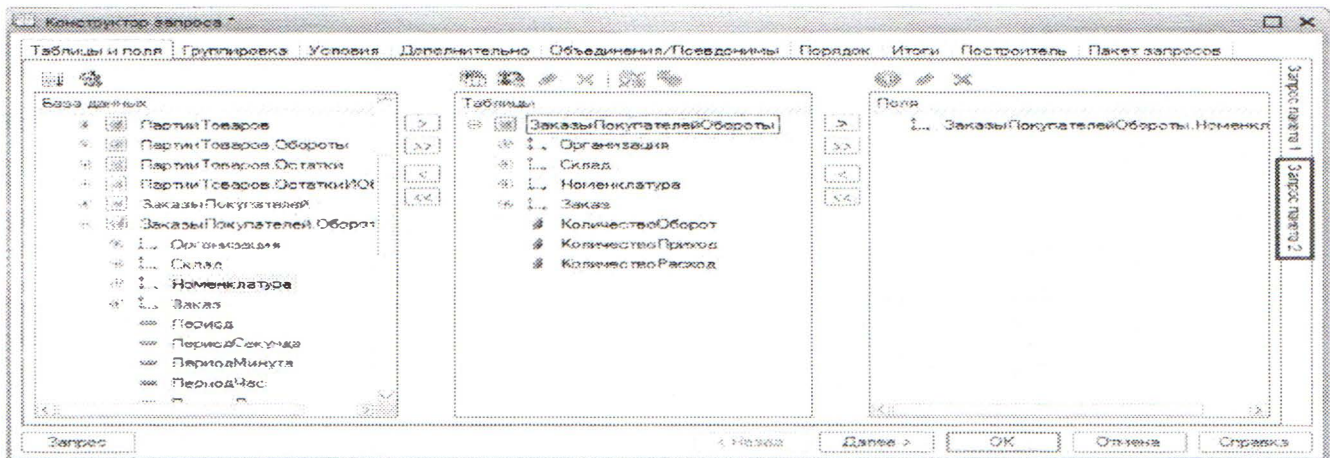


Рисунок 329

На вкладке «Дополнительно» создадим временную таблицу для запроса пакета 2 (Рисунок 330).

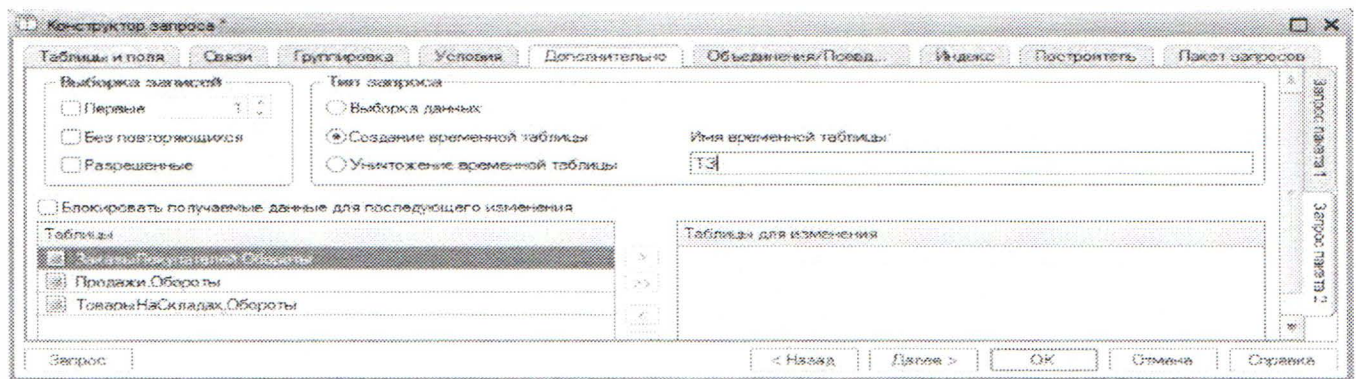


Рисунок 330

После этого запрос пакета 2 будет переименован согласно значению, которое мы указали в поле «Имя временной таблицы» (в данном случае - «ТЗ»), а на вкладке «Таблицы и поля» в запросе пакета 2 отобразилась новая временная таблица, с полями которой можно работать (Рисунок 331).

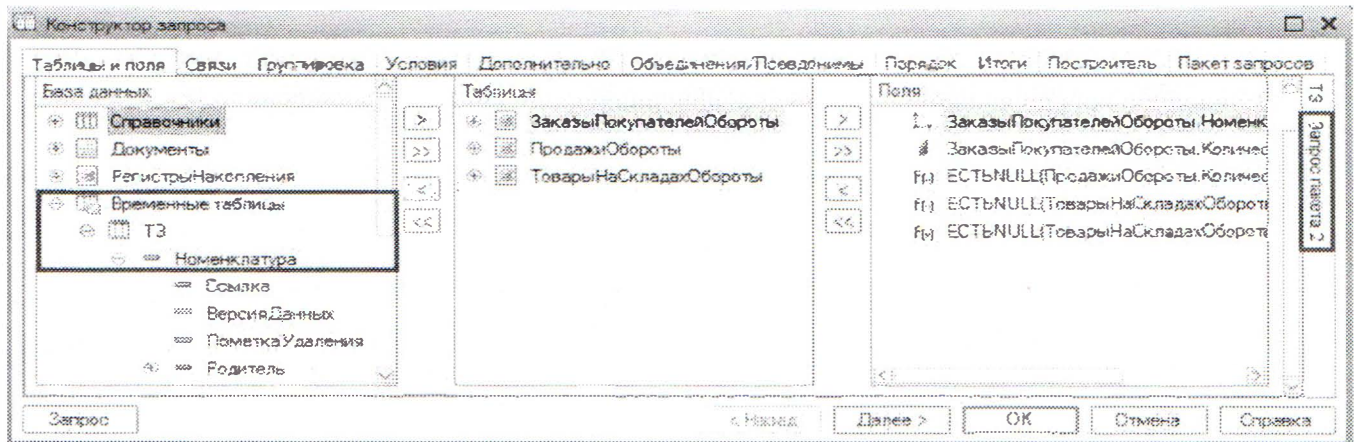


Рисунок 331

Для использования таблицы пакета «ТЗ» заменим текст условия параметров таблиц «ПродажиОбороты» и «ТоварыНаСкладахОбороты» с условием (Рисунок 332).

Номенклатура В

(ВЫБРАТЬ

ИЗ

на следующий:

Номенклатура В

(ВЫБРАТЬ

ТЗ.Номенклатура

ИЗ

ТЗ КАК ТЗ)

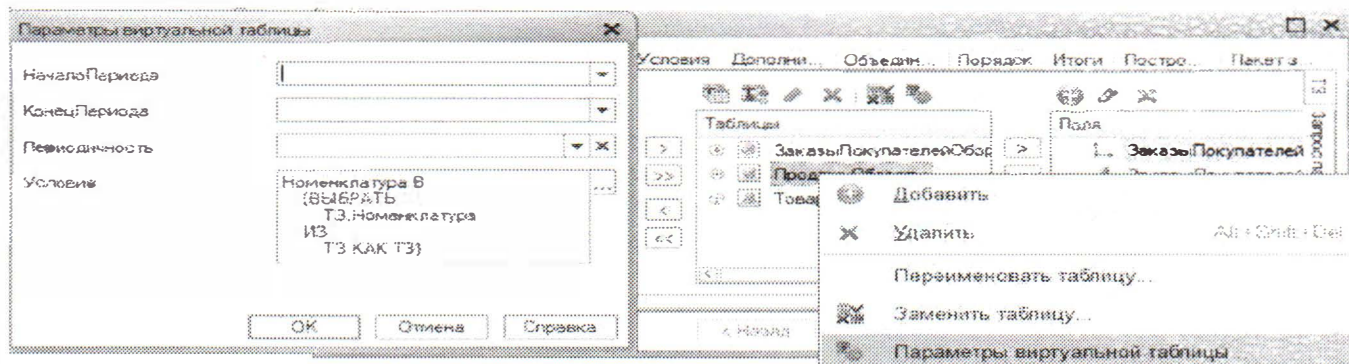


Рисунок 332

В результате выполнения запроса получим запись (Рисунок 333).

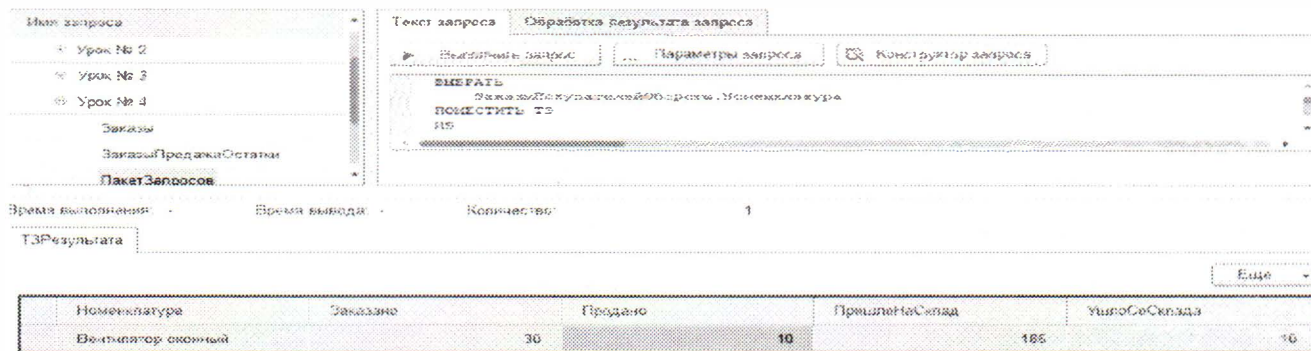


Рисунок 333

Рассмотрим преимущества использования пакетных запросов перед вложенными запросами. Следует отметить, что возможность использования пакетных запросов была добавлена в платформе «1С:Предприятие» позднее, чем вложенные запросы и такая возможность частично отменяет их механизм, так как является более удобной конструкцией (например, помогает избегать добавления вложенных таблиц и их связывания).

В качестве практического примера предлагается решить задачу из теории по вложенным запросам с использованием пакетных запросов.

Для этого выполним действия:

1. Создадим новый запрос «ПакетЗапросовДЗ_3_1» в консоли и выберем поля из виртуальной таблицы оборотов регистра «ПартииТоваров» (Рисунок 334):

ВЫБРАТЬ

ПартииТоваров(Обороты.Склад,

ПартииТоваровОбороты.Номенклатура,

*ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход*

ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

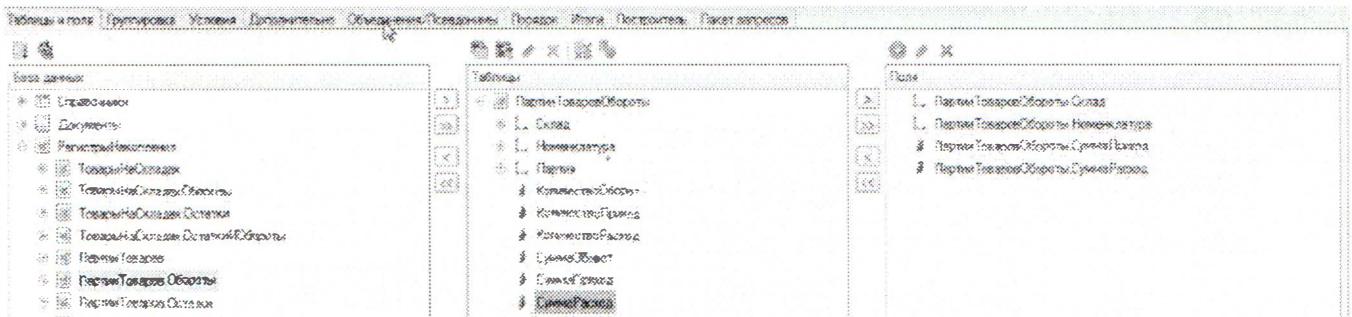


Рисунок 334

2. На закладке «Объединения/Псевдонимы» добавим еще один запрос к виртуальной таблице оборотов регистра «ТоварыНаСкладахОбороты», обратившись к тем же полям (Рисунок 335-336).

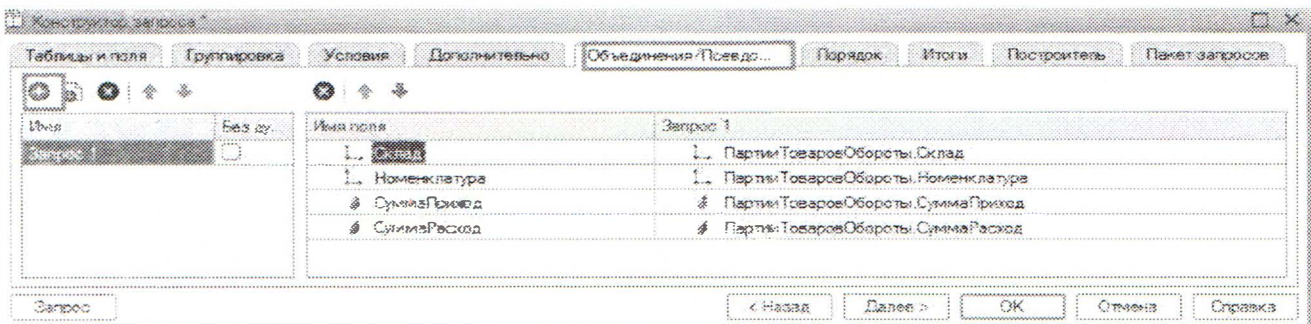


Рисунок 335

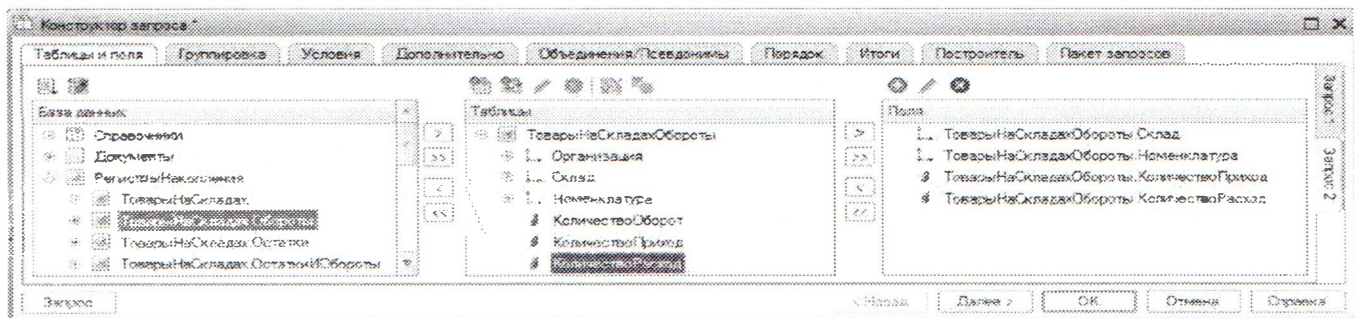


Рисунок 336

Текст объединения запросов выглядит следующим образом:

ВЫБРАТЬ

*ПартииТоваровОбороты.Склад,
ПартииТоваровОбороты.Номенклатура,
ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход,
NULL КАК КоличествоПриход,
NULL КАК КоличествоРасход*

ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

ТоварыНаСкладахОбороты.Склад,

ТоварыНаСкладахОбороты.Номенклатура,
 NULL,
 NULL,
 ТоварыНаСкладахОбороты.КоличествоПриход,
 ТоварыНаСкладахОбороты.КоличествоРасход

ИЗ

РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты

Как из самого текста запроса, так и из конструктора запросов (на закладке «Объединения/Псевдонимы»), можно увидеть, что не все реквизиты запросов пакета связаны (Рисунок 337).

Имя поля	Запрос 1	Запрос 2
Склад	ПартииТоваровОбороты.Склад	ТоварыНаСкладахОбороты.Склад
Номенклатура	ПартииТоваровОбороты.Номенклатура	ТоварыНаСкладахОбороты.Номенклатура
СуммаПриход	ПартииТоваровОбороты.СуммаПриход	«Отсутствует»
СуммаРасход	ПартииТоваровОбороты.СуммаРасход	«Отсутствует»
КоличествоПриход	«Отсутствует»	ТоварыНаСкладахОбороты.КоличествоПриход
КоличествоРасход	«Отсутствует»	ТоварыНаСкладахОбороты.КоличествоРасход

Рисунок 337

Для их связывания добавим недостающие поля в каждом запросе объединения и преобразуем их к «0» (Рисунок 338):

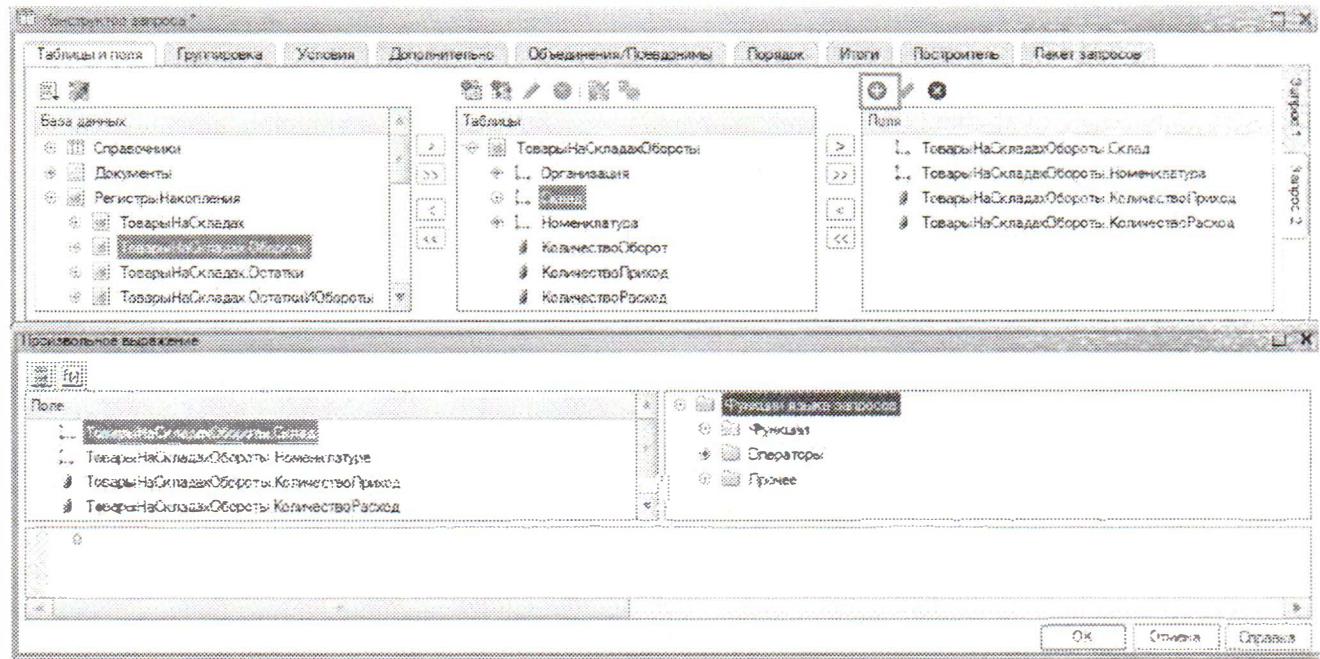


Рисунок 338

Теперь на закладке «Объединения/Псевдонимы» увидим дополнительные поля со значением «0», которые нужно переименовать в поля с отсутствующим значением таким образом, чтобы получилось как на рисунке 339.

Имя	Как зв...	Имя поля	Запрос 1	Запрос 2
Запрос 1		Номенклатура	ПартииТоваровОбороты.Склад	ТоварыНаСкладахОбороты.Склад
Запрос 2		СуммаПриход	ПартииТоваровОбороты.Номенклатура	ТоварыНаСкладахОбороты.Номенклатура
		СуммаРасход	ПартииТоваровОбороты.СуммаПриход	0
		КоличествоПриход	ПартииТоваровОбороты.СуммаРасход	0
		КоличествоРасход	0	ТоварыНаСкладахОбороты.КоличествоПриход
			0	ТоварыНаСкладахОбороты.КоличествоРасход

Рисунок 339

Текст запроса будет выглядеть следующим образом:

ВЫБРАТЬ

*ПартииТоваровОбороты.Склад,
ПартииТоваровОбороты.Номенклатура,
ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход,
0 КАК КоличествоПриход,
0 КАК КоличествоРасход*

ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

*ТоварыНаСкладахОбороты.Склад,
ТоварыНаСкладахОбороты.Номенклатура,
0,
0,
ТоварыНаСкладахОбороты.КоличествоПриход,
ТоварыНаСкладахОбороты.КоличествоРасход*

ИЗ

РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты;

Для проверки результата выполним запрос (Рисунок 340).

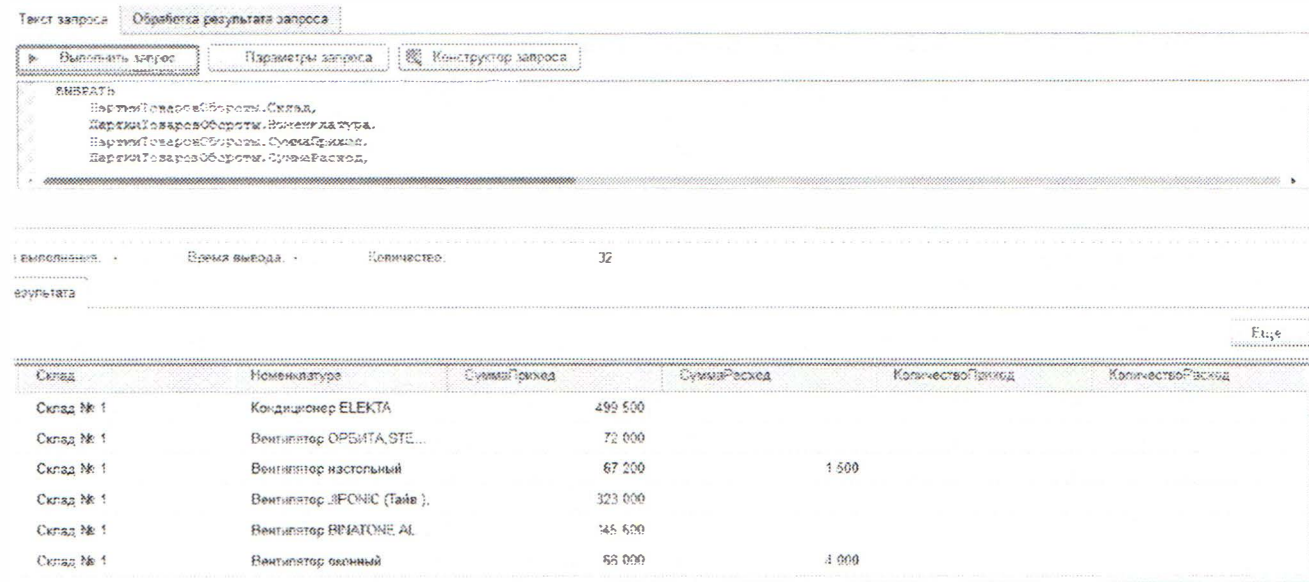


Рисунок 340

Для удобства анализа выгрузим результат запроса в список, на котором видно, что результаты не сгруппированы (Рисунок 341).

Склад	Номенклатура	СуммаПрода	СуммаПокуп	КоличествоПрода	КоличествоПокуп
Склад № 1	Кондиционер ELEKTA	499 000	0	0	0
Склад № 1	Телевизор ORBITA STERLING JR	72 000	0	0	0
Склад № 1	Вентилятор настольный	67 000	1 000	0	0
Склад № 1	Вентилятор (PIONEER Table)	320 000	0	0	0
Склад № 1	Вентилятор BRATONE ALPINE 160cm	149 000	0	0	0
Склад № 1	Вентилятор (PIONEER Table)	88 000	6 000	0	0
Склад № 2	Кондиционер ELEKTA	80 000	0	0	0
Склад № 2	Вентилятор настольный	12 000	0	0	0
Склад № 2	Вентилятор (PIONEER Table)	50 000	0	0	0
Склад № 2	Вентилятор (PIONEER Table)	8 000	0	0	0
Склад № 2	Вентилятор BRATONE ALPINE 160cm	120 000	0	0	0
Склад № 2	Вентилятор ORBITA STERLING JR	40 000	0	0	0
Склад № 2	Вентилятор настольный	1 000	0	0	0
Склад № 1	Стиральная	484 000	0	0	0
Склад № 1	Телевизор "LUC"	150 000	75 000	0	0
Склад № 1	Телевизор "SHARP"	348 000	200 000	0	0
Склад № 1	Вентилятор настольный	0	0	20	0
Склад № 1	Кондиционер ELEKTA	0	0	11	0
Склад № 1	Вентилятор (PIONEER Table)	0	0	170	0
Склад № 1	Вентилятор ORBITA STERLING JR	0	0	100	0
Склад № 1	Вентилятор BRATONE ALPINE 160cm	0	0	112	0
Склад № 2	Вентилятор настольный	0	0	40	0
Склад № 2	Вентилятор (PIONEER Table)	0	0	15	0
Склад № 2	Кондиционер ELEKTA	0	0	20	0
Склад № 2	Вентилятор ORBITA STERLING JR	0	0	100	0
Склад № 2	Вентилятор BRATONE ALPINE 160cm	0	0	94	0
Склад № 2	Вентилятор (PIONEER Table)	0	0	202	0
Склад № 2	Вентилятор настольный	0	0	15	0
Склад № 1	Телевизор "SHARP"	0	0	15	0
Склад № 1	Стиральная	0	0	22	0
Склад № 1	Телевизор "LUC"	0	0	10	0

Рисунок 341

Сгруппируем строки запроса с помощью пакетов запроса. Для этого снова перейдем в конструктор и добавим «Запрос пакета 2» на вкладке «Пакет запросов» (Рисунок 342).

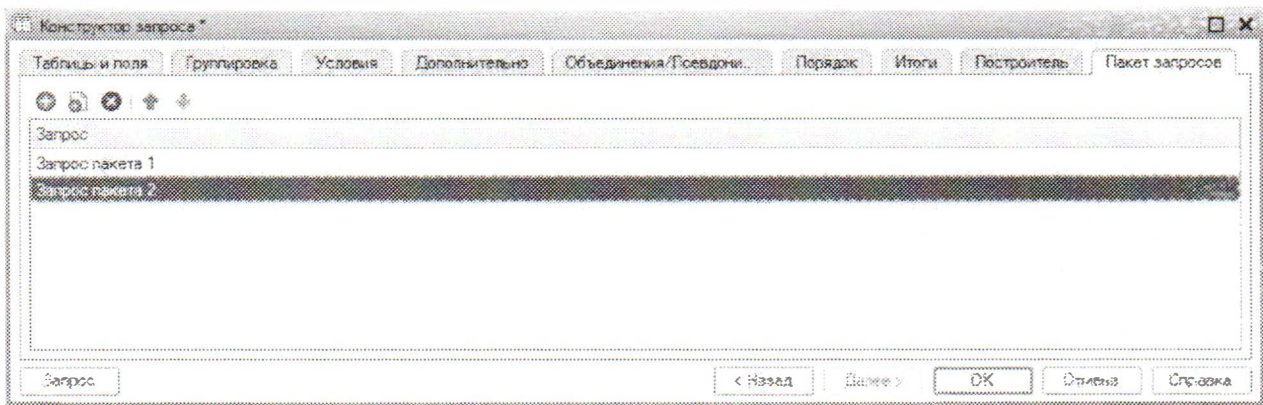


Рисунок 342

Далее поместим во временную таблицу текст предыдущего запроса (назовем ее «ПартииТовары»), чтобы к ней можно было обратиться из «Запроса пакета 2» (Рисунок 343).

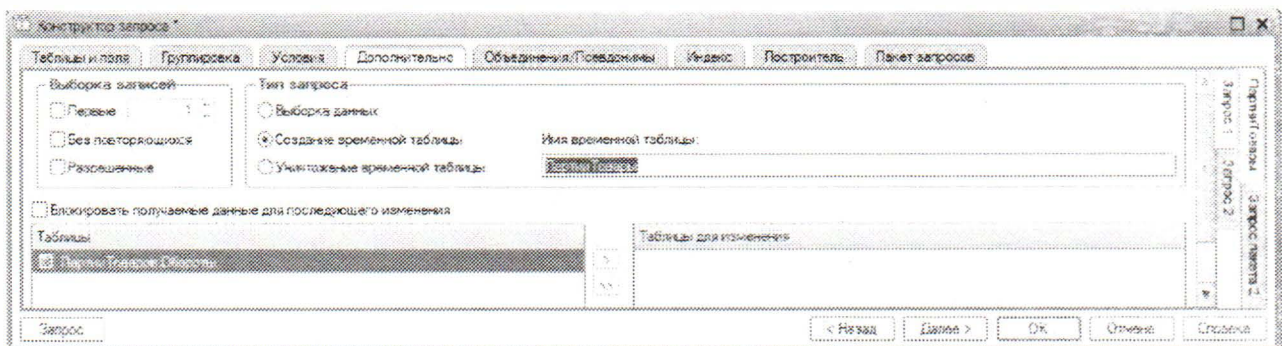


Рисунок 343

Действительно, на вкладке «Таблицы и поля» запроса пакета 2 для выбора стала доступна временная таблица «ПартииТовары», из которой выберем все поля (Рисунок 344).

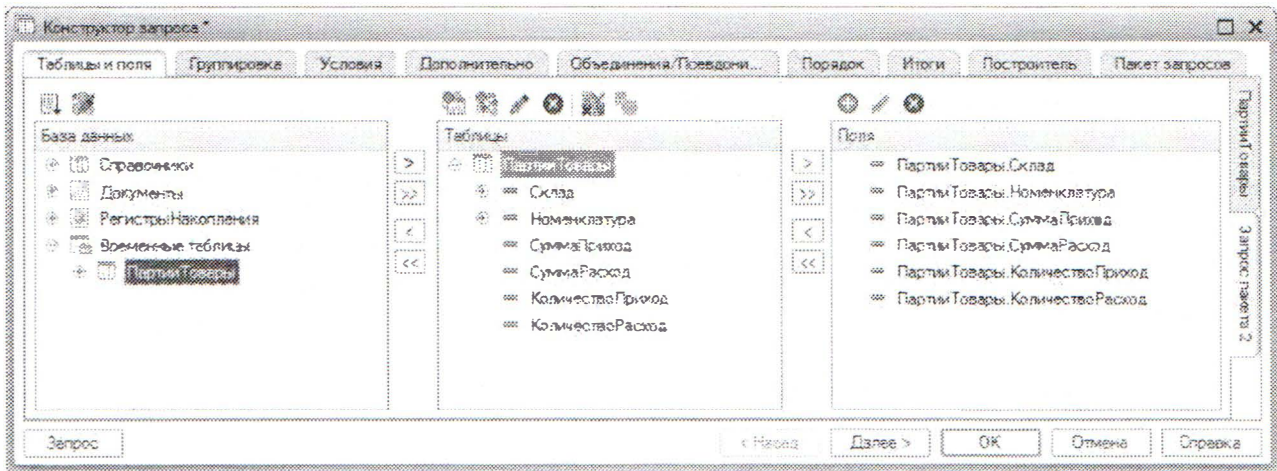


Рисунок 344

Далее в запросе пакета 2 поля со всеми данными сгруппируем по номенклатуре и складу и просуммируем ресурсы (Рисунок 345).

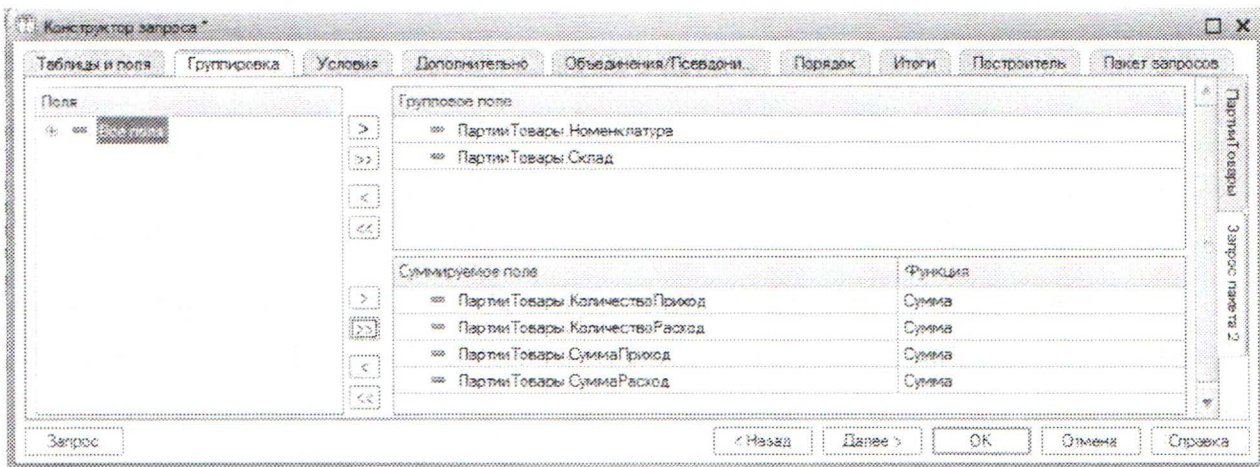


Рисунок 345

Текст запроса после этих действий будет выглядеть следующим образом:

ВЫБРАТЬ

*ПартииТоваровОбороты.Склад,
ПартииТоваровОбороты.Номенклатура,
ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход,
0 КАК КоличествоПриход,
0 КАК КоличествоРасход*

ПОМЕСТИТЬ ПартииТовары

ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

*ТоварыНаСкладахОбороты.Склад,
ТоварыНаСкладахОбороты.Номенклатура,
0,
0,
ТоварыНаСкладахОбороты.КоличествоПриход,
ТоварыНаСкладахОбороты.КоличествоРасход*

ИЗ

РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты;

////////////////////////////////////

ВЫБРАТЬ

ПартииТовары.Склад,
 ПартииТовары.Номенклатура,
 СУММА(ПартииТовары.СуммаПриход) КАК СуммаПриход,
 СУММА(ПартииТовары.СуммаРасход) КАК СуммаРасход,
 СУММА(ПартииТовары.КоличествоПриход) КАК КоличествоПриход,
 СУММА(ПартииТовары.КоличествоРасход) КАК КоличествоРасход

ИЗ

ПартииТовары КАК ПартииТовары
 СГРУППИРОВАТЬ ПО

ПартииТовары.Склад,
 ПартииТовары.Номенклатура

Для проверки корректности запроса проанализируем результат выполнения. После нажатия «Выполнить» в консоли запросов, увидим сгруппированный результат (Рисунок 345).

Склад	Номенклатура	СуммаПриход	СуммаРасход	КоличествоПриход	КоличествоРасход
Склад № 1	Индукционер ELEKTA	436 500	0	0	0
Склад № 1	Вентилятор ORBITA STERLING AP	77 000	0	0	180
Склад № 1	Вентилятор настольный	67 000	0	0	0
Склад № 1	Вентилятор BRUNO ALPINE 100ET	320 000	0	0	170
Склад № 1	Вентилятор BRUNO ALPINE 100ET	145 000	0	0	110
Склад № 1	Вентилятор настольный	68 000	4 000	0	170
Склад № 2	Индукционер ELEKTA	69 000	0	0	0
Склад № 2	Вентилятор настольный	12 000	0	0	40
Склад № 2	Вентилятор BRUNO ALPINE 100ET	60 000	0	0	30
Склад № 2	Вентилятор настольный	8 000	0	0	18
Склад № 2	Вентилятор BRUNO ALPINE 100ET	122 000	0	0	84
Склад № 2	Вентилятор BRUNO ALPINE 100ET	40 000	0	0	300
Склад № 2	Вентилятор настольный	1 500	0	0	5
Склад № 1	Термометр JVC	48 000	0	0	0
Склад № 1	Термометр JVC	140 000	75 000	0	10
Склад № 1	Термометр JVC	345 000	730 000	0	10

Рисунок 345

Таким образом, мы реализовали задачу с использованием пакета запросов. В сравнении с вложенными запросами, такая конструкция является более компактной (в случае с вложенными запросами мы бы видели дерево таблиц в консоли запросов, что замедляло бы его восприятие). Из запроса пакета «Партии товаров» мы сразу переходим к подзапросам объединения. Таким образом, пакеты запросов в практике использования являются более удобными, чем вложенные запросы, однако, в качестве исключения, следует назвать ситуации, когда запросов пакета очень много. В таком случае, для удобства восприятия, также можно использовать вложенные запросы.

Рассмотрим расширенные возможности пакетных запросов.

Для этого создадим в конфигураторе отчет «Пакет запросов» со стандартными реквизитом формы «ТабДок» и командой «Сформировать» (Рисунок 346):

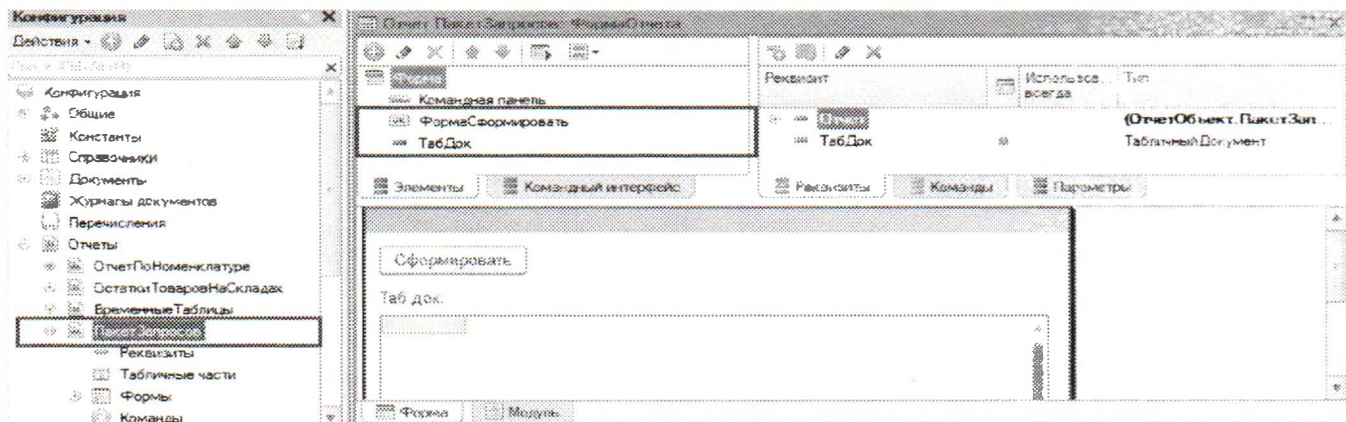


Рисунок 346

В процедуру «СформироватьНаСервере(ТабДок)» добавим текст запроса предыдущей

главы:

ВЫБРАТЬ

*ПартииТоваровОбороты.Склад,
ПартииТоваровОбороты.Номенклатура,
ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход,
0 КАК КоличествоПриход,
0 КАК КоличествоРасход*

ПОМЕСТИТЬ ПартииТовары

ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

*ТоварыНаСкладахОбороты.Склад,
ТоварыНаСкладахОбороты.Номенклатура,
0,
0,
ТоварыНаСкладахОбороты.КоличествоПриход,
ТоварыНаСкладахОбороты.КоличествоРасход*

ИЗ

РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты:

////////////////////////////////////

ВЫБРАТЬ

*ПартииТовары.Склад,
ПартииТовары.Номенклатура,
СУММА(ПартииТовары.СуммаПриход) КАК СуммаПриход,
СУММА(ПартииТовары.СуммаРасход) КАК СуммаРасход,
СУММА(ПартииТовары.КоличествоПриход) КАК КоличествоПриход,
СУММА(ПартииТовары.КоличествоРасход) КАК КоличествоРасход*

ИЗ

ПартииТовары КАК ПартииТовары

СТРУППИРОВАТЬ ПО

*ПартииТовары.Склад,
ПартииТовары.Номенклатура*

В случае использования метода «ЗапросВыполнить()», отработывает последний «Запрос пакета 2». При этом если «Запрос пакета 2» поместить во временную таблицу и поставить точку останова на методе «Запрос.Выполнить()Выгрузить()», то в отладчике увидим просто «СтрокуТаблицыЗначений» без расшифровки. Это говорит о том, что последним запросом пакета должна быть не временная таблица, а выборка данных. Таким образом, при исполнении пакета запросов мы используем множество таблиц, при соединении воедино которых, получим результат запроса

Рассмотрим механизм исполнения запроса при методе «ВыполнитьПакет()», при котором отработывает не последний в списке запрос пакетов, а последовательно выполняются все запросы, как будто для каждого из них отработал метод «Выполнить()».

При этом результаты запросов помещаются в массив в последовательности расположения запросов в тексте пакета. Для примера выполним следующие действия: создадим «Запрос пакета 3» и добавим из временной таблицы «ПартииТовары» только склад и номенклатуру (Рисунок 347).

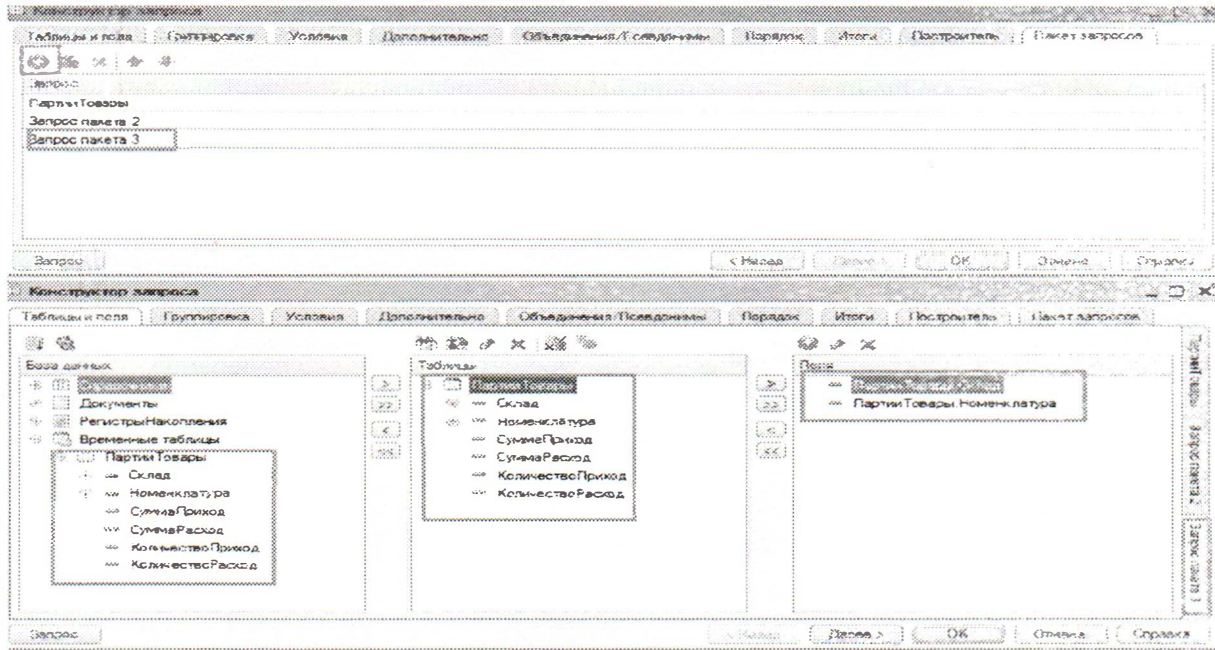


Рисунок 347

Текст запроса:

Запрос2 = Новый Запрос;

Запрос2.Текст = "ВЫБРАТЬ

ПартииТоваровОбороты.Склад,
ПартииТоваровОбороты.Номенклатура,
ПартииТоваровОбороты.СуммаПриход,
ПартииТоваровОбороты.СуммаРасход,
0 КАК КоличествоПриход,
0 КАК КоличествоРасход

ПОМЕСТИТЬ ПартииТовары
ИЗ

РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

ТоварыНаСкладахОбороты.Склад,
ТоварыНаСкладахОбороты.Номенклатура,
0,
0,
ТоварыНаСкладахОбороты.КоличествоПриход,
ТоварыНаСкладахОбороты.КоличествоРасход

ИЗ

РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты

;

ВЫБРАТЬ

ПартииТовары.Склад,


```

ПартииТовары.Номенклатура,
СУММА(ПартииТовары.СуммаПриход) КАК СуммаПриход,
СУММА(ПартииТовары.СуммаРасход) КАК СуммаРасход,
СУММА(ПартииТовары.КоличествоПриход) КАК КоличествоПриход,
СУММА(ПартииТовары.КоличествоРасход) КАК КоличествоРасход
ИЗ
ПартииТовары КАК ПартииТовары

СТРУППИРОВАТЬ ПО
ПартииТовары.Склад,
ПартииТовары.Номенклатура
:
////////////////////////////////////
ВЫБРАТЬ
ПартииТовары.Склад,
ПартииТовары.Номенклатура
ИЗ
ПартииТовары КАК ПартииТовары";
Запрос2.ВыполнитьПакет();

```

Поставим точку останова на строке «Запрос2.ВыполнитьПакет()», запустим приложение под отладкой и посмотрим результат (Рисунок 348).

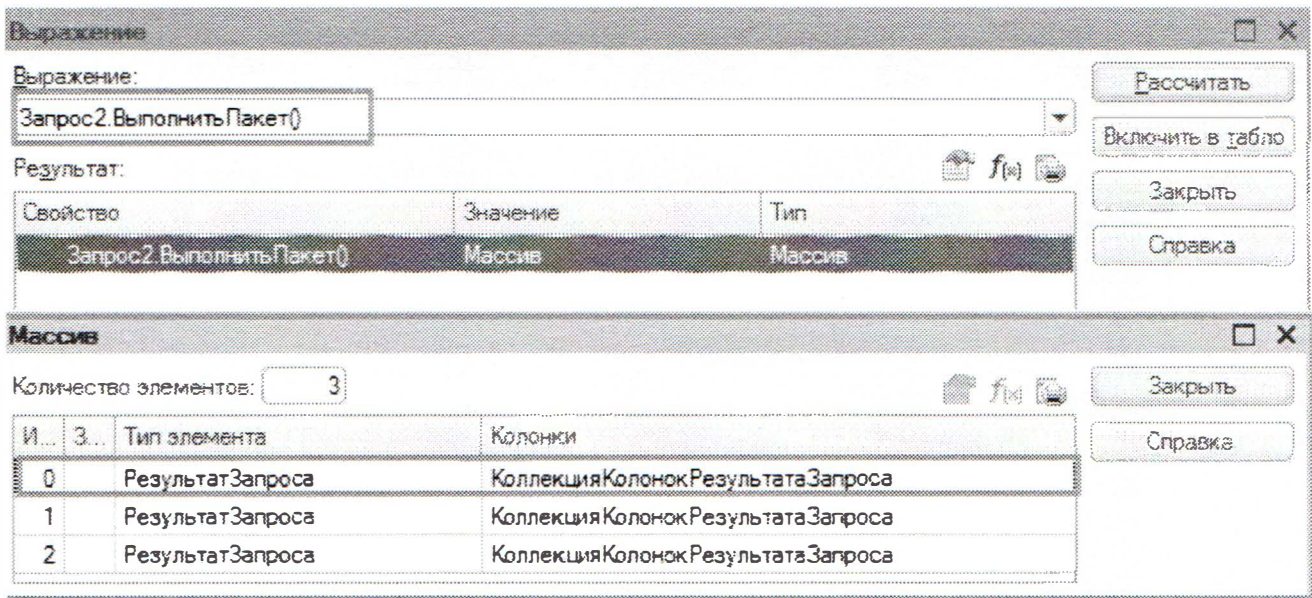


Рисунок 348

В верхней части изображения мы видим результат метода «ВыполнитьПакет()» со значением типа «Массив». Для того чтобы увидеть значения элементов массива результатов запросов пакета, откроем результат в отдельном окне (нижняя часть изображения), где каждый результат запроса соответствует выполнению одного запроса пакета. Нумерация элементов массива начинается с нуля, соответственно, для детализации данных по первому, например, запросу, выполним метод «Запрос 2.Выполнить пакет.()[0]Выгрузить()» (Рисунок 349).

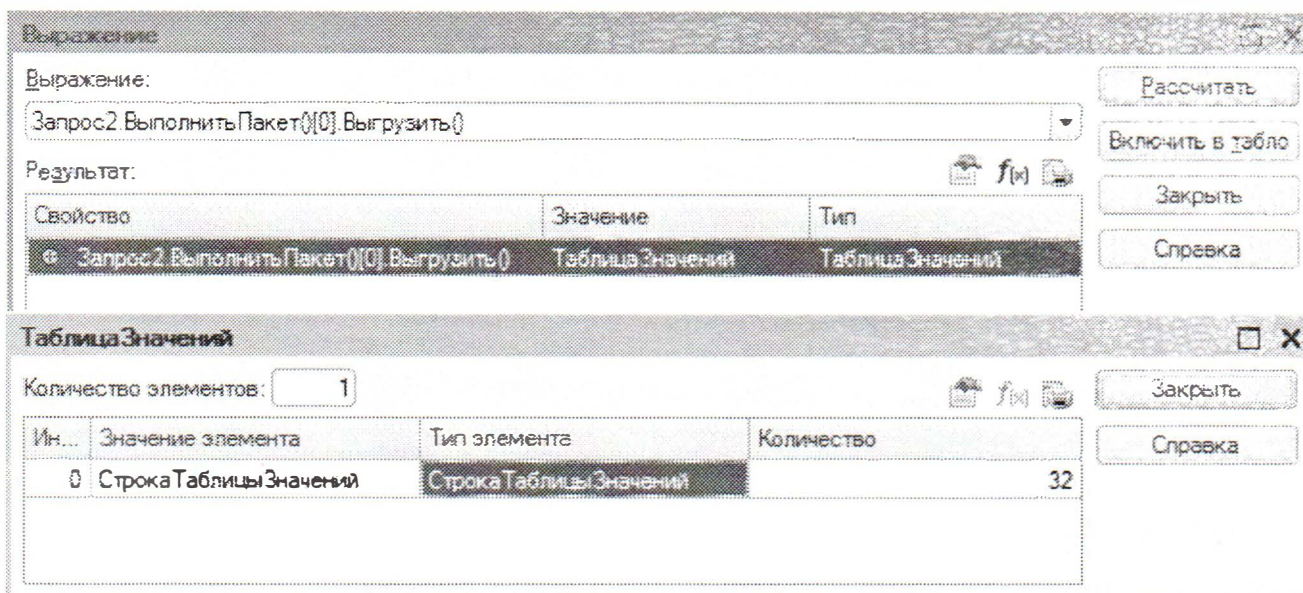


Рисунок 349

Полученные данные не пригодны для анализа, так как представляют собой строку с отображением количества записей без детализации результатов. Связано это с тем, что первый запрос в пакете является временной таблицей. В этом можно убедиться в конструкторе запроса (Рисунок 350).

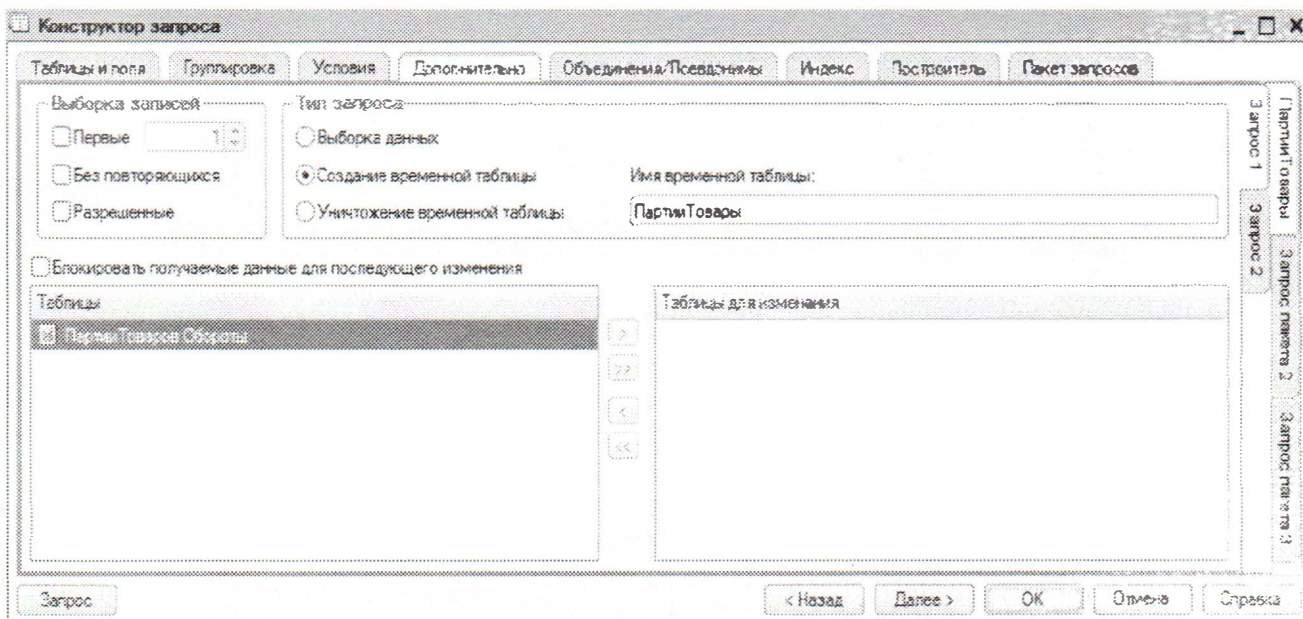


Рисунок 350

Если мы выполним те же действия для следующего запроса в пакете, который является выборкой данных, то получим детализированный до записей результат. Для этого нужно выполнить метод «Запрос2.ВыполнитьПакет()[1].Выгрузить()» и открыть результат выполнения (теперь уже – «ТаблицуЗначений») в отдельном окне, в котором мы увидим все записи (Рисунок 351).

Выражение:
Запрос2.ВыполнитьПакет(1).Выгрузить()

Результат

Свойство	Значение	Тип
Запрос2.ВыполнитьПакет(1).Выгрузить()	Таблица.Значений	Таблица.Значений

Таблица.Значений

Количество элементов: 16

Индекс	Значение элемента	Тип элемента	Склад	Номенклатура	СуммаГривод	СуммаРасход	КоличествоГрив	КоличествоРасход
0	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Кондиционер EL	499 000	0	111	0
1	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Вентилятор OPE...	72 000	0	130	0
2	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Вентилятор нас...	6 7 200	1 500	224	5
3	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Вентилятор JIP...	323 000	0	170	0
4	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Вентилятор BIN...	145 600	0	112	0
5	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Вентилятор око...	68 000	4 000	170	10
6	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Кондиционер EL	90 000	0	20	0
7	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Вентилятор нас...	12 000	0	40	0
8	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Вентилятор JIP...	80 000	0	32	0
9	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Вентилятор око...	5 000	0	15	0
10	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Вентилятор BIN...	122 200	0	34	0
11	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 2	Вентилятор OPE...	40 000	0	130	0
12	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 3	Вентилятор нас...	1 500	0	5	0
13	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	СТИНОП 101	434 000	0	22	0
14	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Телевизор "JVC"	150 000	75 000	10	5
15	СтрокаТаблицы...	СтрокаТаблицы...	Склад № 1	Телевизор "SH	245 000	230 000	15	10

Рисунок 351

Соответственно, результатом выполнения третьего запроса будет таблица с колонками «Склад» и «Номенклатура» (Рисунок 352).

Выражение:
Запрос2.ВыполнитьПакет(2).Выгрузить()

Результат

Свойство	Значение	Тип
Запрос2.ВыполнитьПакет(2).Выгрузить()	Таблица.Значений	Таблица.Значений

Таблица.Значений

Количество элементов: 32

Инд	Значение элемента	Тип элемента	Склад	Номенклатура
0	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Кондиционер ELEKTA
1	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор OPEBITA STERLING.ЯП.
2	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор настольный
3	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор JIPONIC (Taps)
4	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор BINATONE ALPINE 160вт. на.
5	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор оконный
6	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Кондиционер ELEKTA
7	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Вентилятор настольный
8	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Вентилятор JIPONIC (Taps)
9	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Вентилятор оконный
10	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Вентилятор BINATONE ALPINE 160вт. на.
11	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 2	Вентилятор OPEBITA STERLING.ЯП.
12	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 3	Вентилятор настольный
13	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	СТИНОП 101
14	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Телевизор "JVC"
15	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Телевизор "SHARP"
16	СтрокаТаблицы.Значений	СтрокаТаблицы.Значений	Склад № 1	Вентилятор настольный

Рисунок 352

Таким образом, в ходе исполнения пакетного запроса, мы получили данные, которые в обычных условиях можно было бы получить только несколькими запросам, а не одним.

Для удобства использования результатов запросов пакета, передадим результат выполнения пакетов в массив. Для этого заменим последнюю строку запроса (Рисунок 353):

«Запрос2.ВыполнитьПакет()»

на

МассивЗапросов = Запрос2.ВыполнитьПакет();

Выборка1 = МассивЗапросов[1].Выбрать();

Выборка2 = МассивЗапросов[2].Выбрать();

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
: ВЫБРАТЬ
:   ПартииТовары.Склад,
:   ПартииТовары.Номенклатура,
:   СУММА (ПартииТовары.СуммаПриход) КАК СуммаПриход,
:   СУММА (ПартииТовары.СуммаРасход) КАК СуммаРасход,
:   СУММА (ПартииТовары.КоличествоПриход) КАК КоличествоПриход,
:   СУММА (ПартииТовары.КоличествоРасход) КАК КоличествоРасход
: ИЗ
:   ПартииТовары КАК ПартииТовары
:
: СГРУППИРОВАТЬ ПО
:   ПартииТовары.Склад,
:   ПартииТовары.Номенклатура
: ;
:
:////////////////////////////////////////////////////////////////////////////////////////////////////////////////
: ВЫБРАТЬ
:   ПартииТовары.Склад,
:   ПартииТовары.Номенклатура
: ИЗ
:   ПартииТовары КАК ПартииТовары";
МассивЗапросов = Запрос2.ВыполнитьНамет();
Выборка1 = МассивЗапросов[1].Выбрать();
Выборка2 = МассивЗапросов[2].Выбрать();
КонечнаяПроцедура
```

Рисунок 353

Смысл в том, что если выборка позволяет получить несколько результатов запросов, то это можно выполнить в рамках одного пакета запросов.

Отметим, что результатом выполнения запроса на уничтожение временной таблицы, является значение «Неопределенно», которое также помещается в массив результатов. Результатом выполнения запроса на создание временной таблицы будет результат с одной колонкой и одной строкой, содержащей количество записей, помещенных в созданную временную таблицу.

Итоговый текст запроса процедуры «СформироватьНаСервере(ТабДок)» будет иметь вид:

Процедура СформироватьНаСервере(ТабДок)

```
Запрос2 = Новый Запрос;
Запрос2.Текст = "ВЫБРАТЬ
|   ПартииТоваровОбороты.Склад,
|   ПартииТоваровОбороты.Номенклатура,
|   ПартииТоваровОбороты.СуммаПриход,
|   ПартииТоваровОбороты.СуммаРасход,
|   0 КАК КоличествоПриход,
|   0 КАК КоличествоРасход
ПОМЕСТИТЬ ПартииТовары
ИЗ
|   РегистрНакопления.ПартииТоваров.Обороты КАК ПартииТоваровОбороты
ОБЪЕДИНИТЬ ВСЕ
ВЫБРАТЬ
|   ТоварыНаСкладахОбороты.Склад,
|   ТоварыНаСкладахОбороты.Номенклатура,
|   0,
|   0,
|   ТоварыНаСкладахОбороты.КоличествоПриход,
|   ТоварыНаСкладахОбороты.КоличествоРасход
ИЗ
|   РегистрНакопления.ТоварыНаСкладах.Обороты КАК ТоварыНаСкладахОбороты;
```

```

|ВЫБРАТЬ
|    ПартииТовары.Склад,
|    ПартииТовары.Номенклатура,
|    СУММА(ПартииТовары.СуммаПриход) КАК СуммаПриход,
|    СУММА(ПартииТовары.СуммаРасход) КАК СуммаРасход,
|    СУММА(ПартииТовары.КоличествоПриход) КАК КоличествоПриход,
|    СУММА(ПартииТовары.КоличествоРасход) КАК КоличествоРасход
|ИЗ
|    ПартииТовары КАК ПартииТовары
|
|СГРУППИРОВАТЬ ПО
|    ПартииТовары.Склад,
|    ПартииТовары.Номенклатура;
|////////////////////////////////////
|ВЫБРАТЬ
|    ПартииТовары.Склад,
|    ПартииТовары.Номенклатура
|ИЗ
|    ПартииТовары КАК ПартииТовары";
МассиЗапросов = Запрос2.ВыполнитьПакет();
Выборка1 = МассиЗапросов[1].Выбрать();
Выборка2 = МассиЗапросов[2].Выбрать();
КонецПроцедуры

```

Глава 5.

5.1. Оперативный учет.

5.1.1. Ввод в эксплуатацию.

В предыдущих главах был рассмотрен теоретический материал по использованию языка запросов 1С. В данной главе предлагается перейти к рассмотрению параметров запросов и структуры таблиц на примере решения практических задач, поскольку примеры, рассмотренные ранее, являются простыми и не дают полноты картины использования механизма запросов.

В качестве практикума, разберем задачу из курса подготовки к квалификации «Специалист по платформе «1С:Предприятие:8.3»», которая предоставляет минимум исходных данных, приближенных к ситуации реальной автоматизации. Следует отметить, что задачей данной главы не является подготовка к экзамену, поэтому акцентировать внимание на логику построения структуры метаданных мы не будем, наша цель – в результате решения данных задач, проанализировать запросы.

Итак, формулировка задачи выглядит следующим образом (Рисунок 354).

В компании используется оборудование, имеющее ограничения к эксплуатации. Для каждого устройства устанавливается срок годности и срок использования. Срок годности считается с момента приобретения оборудования и указывается в документе поступления, а срок эксплуатации с момента его ввода в эксплуатацию и указывается непосредственно для оборудования и не меняется. Поступление оборудования отражается документом «Приходная накладная». Через некоторое время оборудование вводится в эксплуатацию документом «Ввод в эксплуатацию».

При передаче оборудования в эксплуатацию в первую очередь передается оборудование, у которого минимальный срок годности. В документе «Ввод в эксплуатацию» указывается перечень и количество передаваемого оборудования. При проведении должна происходить проверка на наличие оборудования и его срок годности. В том случае, если срок годности истек или оборудования недостаточно, документ не проводится и выдается соответствующее сообщение.

Раз в месяц формируется регламентный документ «Выбытие оборудования», который при проведении проверяет эксплуатируемое оборудование и, если оно негодно, то списывает его. При проверке определяется, вышел ли срок эксплуатации устройства с момента его ввода в эксплуатацию. Если срок эксплуатации истек или истек срок годности, то оборудование должно быть списано. Кроме того, этот же документ должен списывать еще не введенное в эксплуатацию оборудование, но срок годности, которого уже истек.

Учет оборудования в разрезе складов не ведется. Себестоимость оборудования рассчитывается как средняя.

Необходимо создать отчет о состоянии эксплуатируемого оборудования на выбранную дату.

Состояние оборудования в эксплуатации на 31.01.2010

Оборудование	Кол-во	Сумма	Оставшийся срок годв.	Оставшийся срок эксп.
Киникамера	3	3 000	60	20
Портсигар	2	400	500	520

Рисунок 354

Рассмотрим структуру метаданных конфигурации, созданную для решения этой задачи. Скачать ее можно по ссылке <https://yadi.sk/d/3rE3dCapdvrF1Q>.

В дереве метаданных видим следующие объекты:

– Справочник «Номенклатура» хранит данные по оборудованию в информационной базе.

Реквизиты (Рисунок 355):

1. СрокЭксплуатации (тип значения: Число) – срок эксплуатации, который задается для номенклатурной позиции;
2. Комплект (тип значения: Булево) определяет, входит ли номенклатура в состав комплекта.

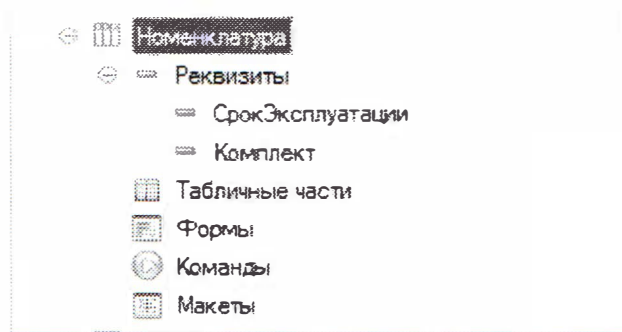


Рисунок 355

– Документ «Приходная накладная» регистрирует приход оборудования, которое фиксируется в табличной части документа «Список номенклатуры» с указанием срока годности (для очередности ввода в эксплуатацию), количества и цены (для расчета себестоимости) (Рисунок 356).

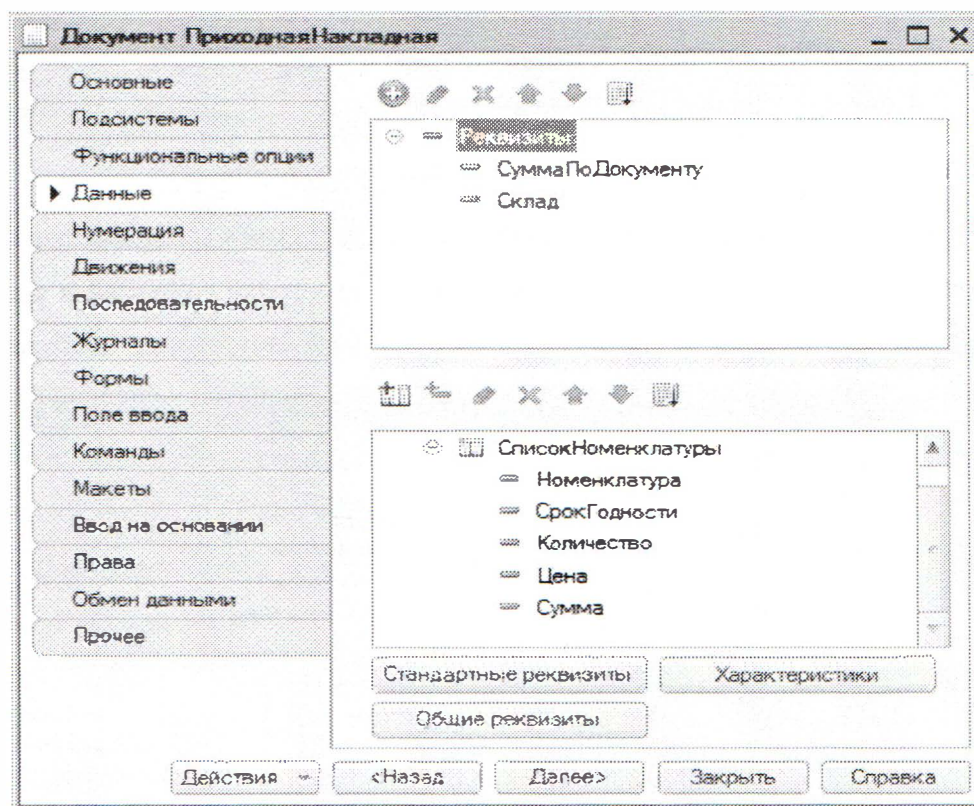


Рисунок 356

Документ делает движения по регистрам: «Остатки номенклатуры», «Оборудование в эксплуатации» и бухгалтерскому регистру «Управленческий» (Рисунок 357).

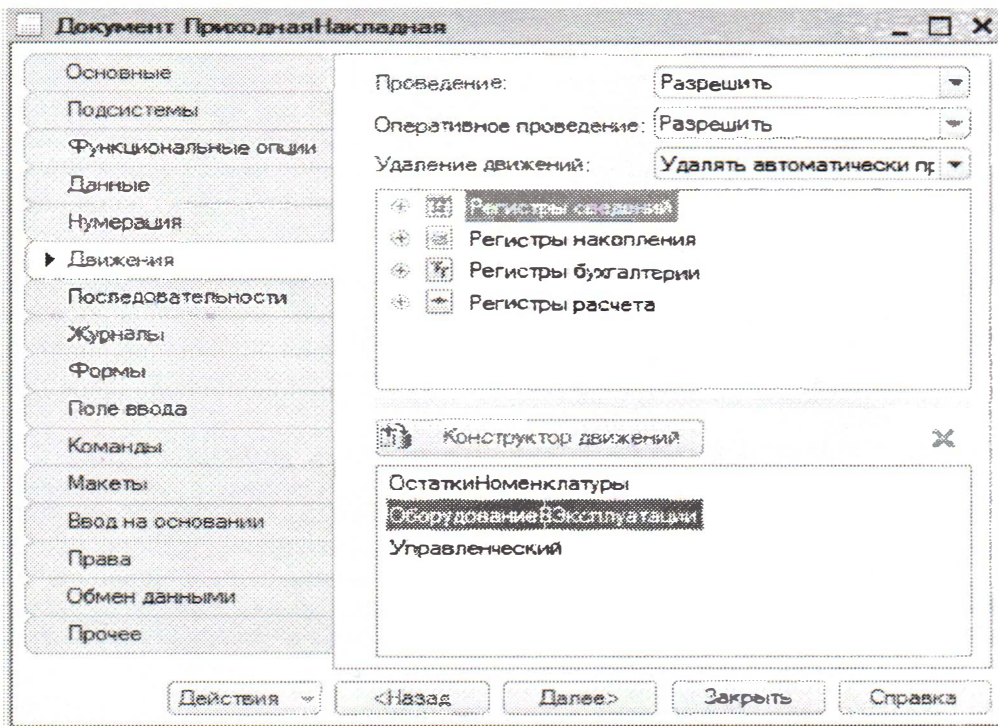


Рисунок 357

Регистр накопления «Оборудование в эксплуатации» содержит записи факта ввода в эксплуатацию и выбытия оборудования с учетом его срока годности и срока эксплуатации (Рисунок 358).

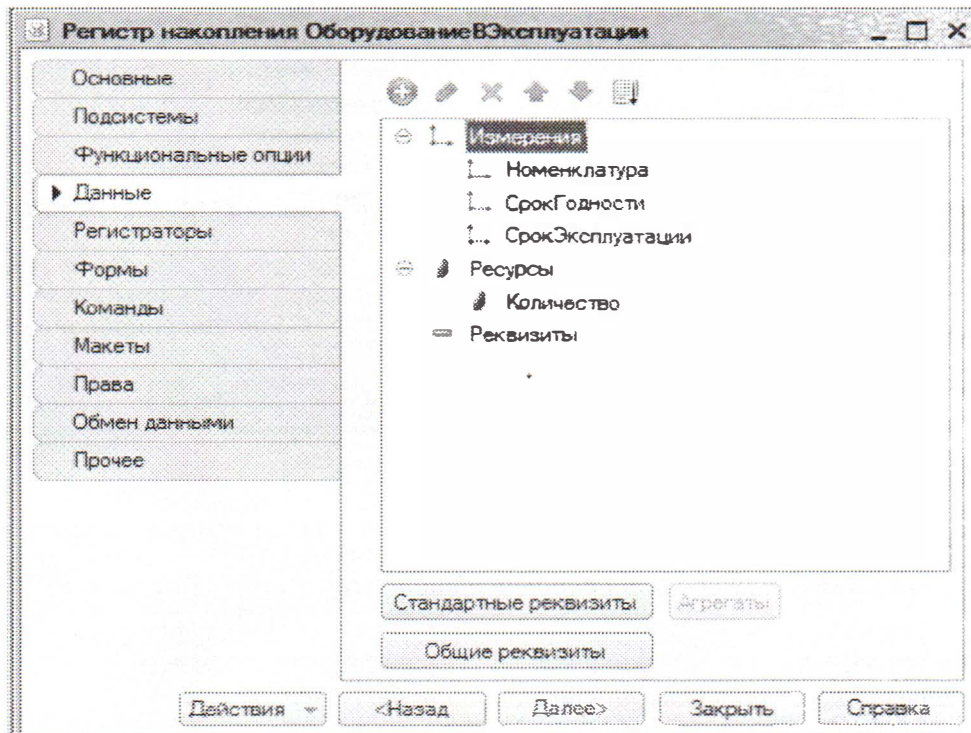


Рисунок 358

Подчинен следующим регистраторам (Рисунок 359).

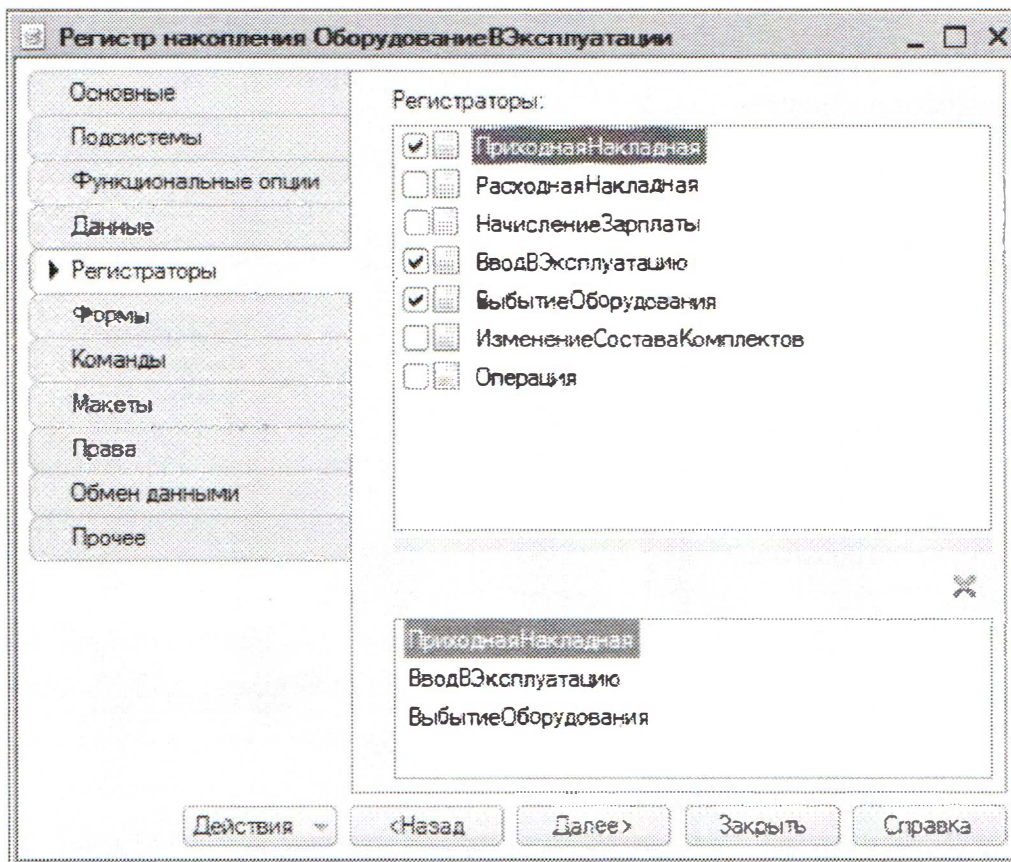


Рисунок 359

Особенности заполнения записей регистра таковы: в момент прихода оборудования срока эксплуатации еще нет, заполнены реквизиты: «Номенклатура», «Количество» и «Срок годности». Поле «Срок эксплуатации» заполняется после регистрации ввода в эксплуатацию оборудования документом «Ввод в эксплуатацию»:

- Документ «Ввод в эксплуатацию» регистрирует факт ввода оборудования в эксплуатацию (Рисунок 360).

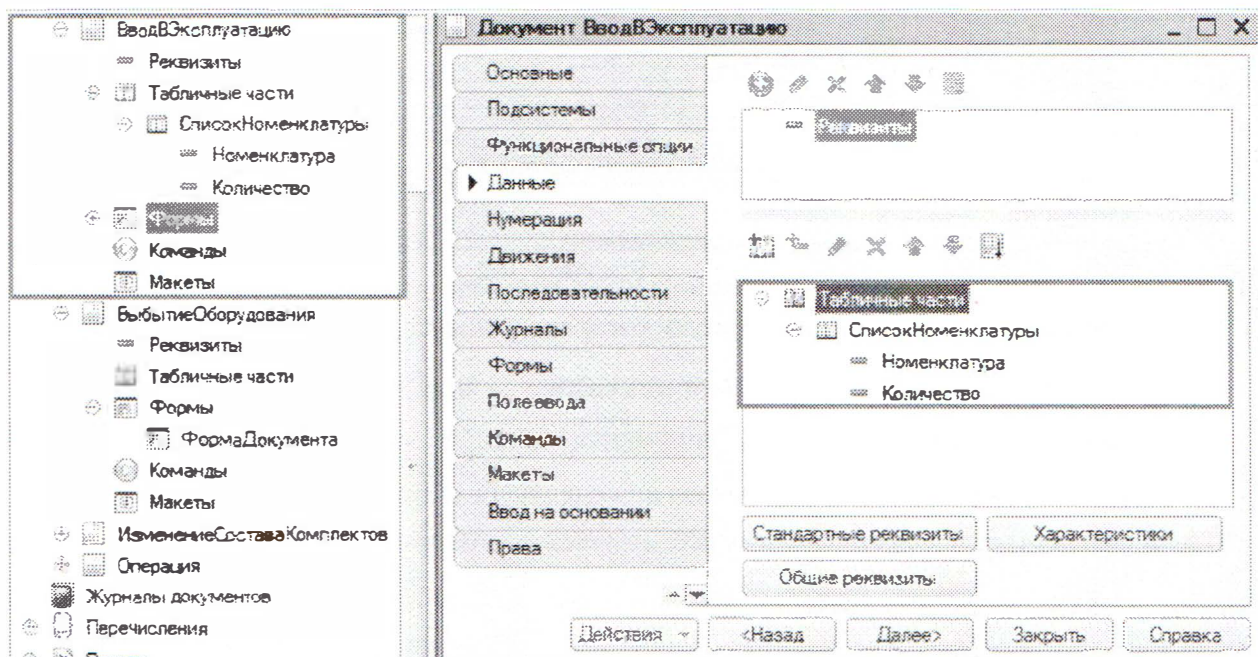


Рисунок 360

– Документ «Выбытие оборудование» регистрирует факт списания оборудования из эксплуатации (Рисунок 361).

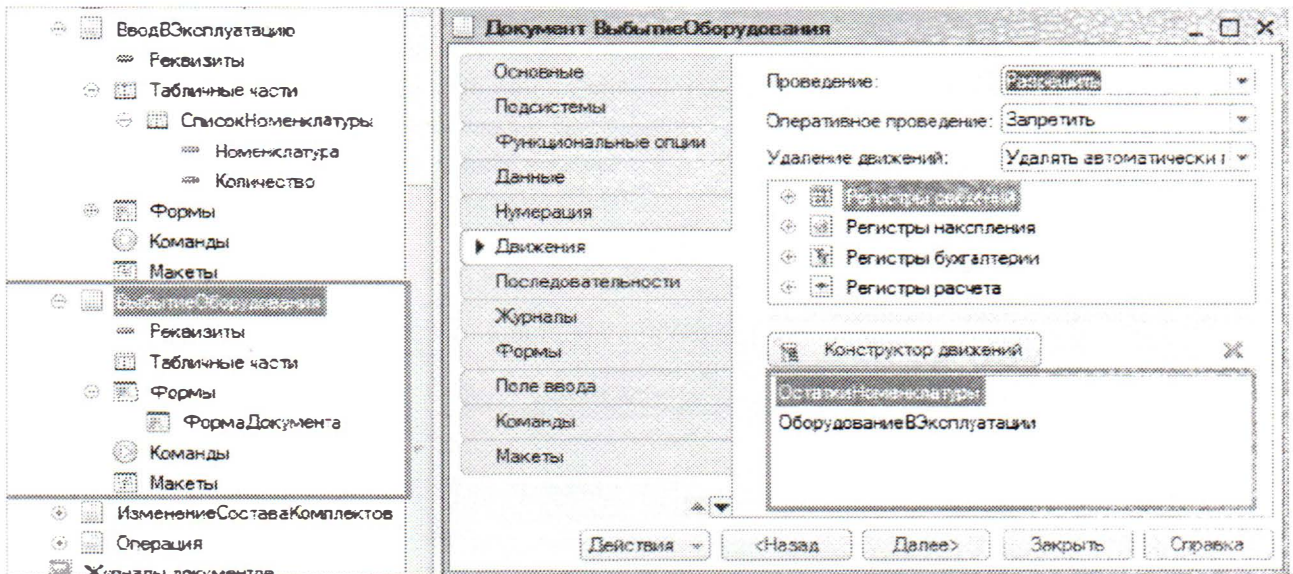


Рисунок 361

Теперь рассмотрим запросы.

В приходной накладной запросов нет, движения документа выглядят следующим образом (Рисунок 362).

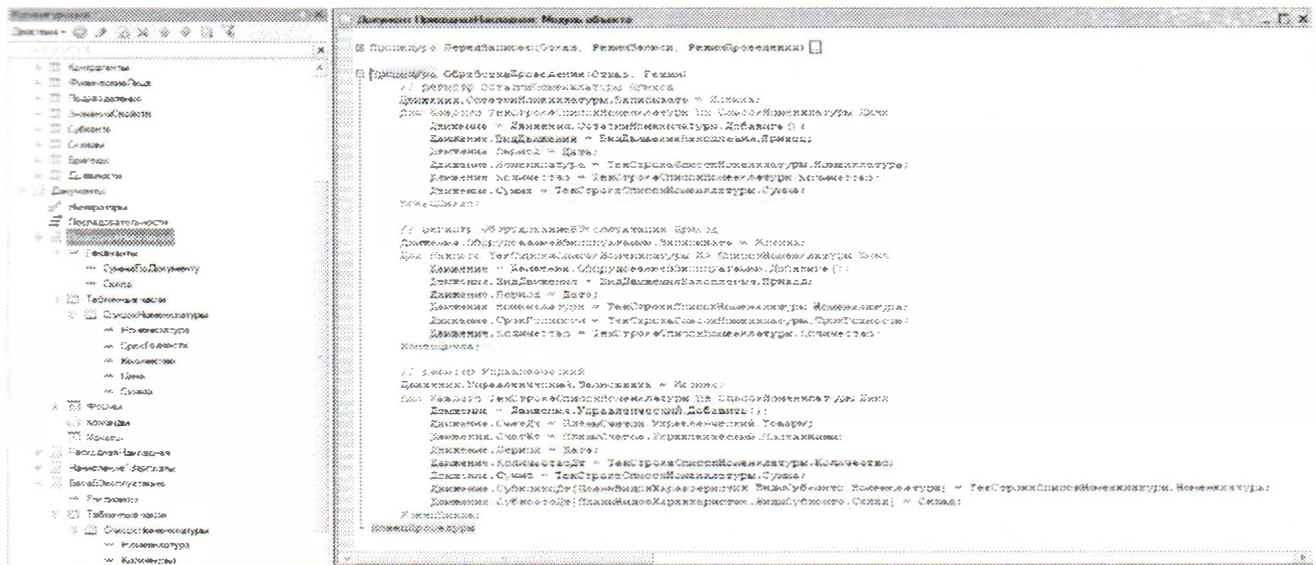


Рисунок 362

Рассмотрим запрос, с помощью которого решается задача передачи в эксплуатацию оборудования с минимальным сроком годности.

Для этого откроем форму документа «Ввод в эксплуатацию», в табличной части которой отображены только данные по номенклатуре и количеству (Рисунок 363).

Рисунок 363

Необходимо реализовать проверку достаточности данного количества на остатках в момент ввода оборудования, и вводить в эксплуатацию номенклатуру с минимальным сроком годности.

Перейдем в модуль объекта документа и проанализируем обработку проведения:

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОборудованиеВЭксплуатации.БлокироватьДляИзменения = Истина;

Движения.ОборудованиеВЭксплуатации.Записывать = Истина;

Движения.ОборудованиеВЭксплуатации.Записать();

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки =

Блокировка.Добавить("РегистрНакопления.ОборудованиеВЭксплуатации");

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = СписокНоменклатуры;

*ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");*

Блокировка.Заблокировать();

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ВводВЭксплуатациюСписокНоменклатуры.Номенклатура КАК

Номенклатура,

| СУММА(ВводВЭксплуатациюСписокНоменклатуры.Количество) КАК

Количество,

|

ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации КАК

СрокЭксплуатации

|ПОМЕСТИТЬ ДанныеТЧ

|ИЗ

| *Документ.ВводВЭксплуатацию.СписокНоменклатуры КАК*

ВводВЭксплуатациюСписокНоменклатуры

| **ГДЕ**

| *ВводВЭксплуатациюСписокНоменклатуры.Ссылка = &Ссылка*

| **СГРУППИРОВАТЬ ПО**

| *ВводВЭксплуатациюСписокНоменклатуры.Номенклатура,*

ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации

| **ИНДЕКСИРОВАТЬ ПО**

| *Номенклатура*

| ;

| //

| **ВЫБРАТЬ**

| *ДанныеТЧ.Номенклатура КАК Номенклатура,*

| *ДанныеТЧ.Количество КАК Количество,*

| *ДанныеТЧ.СрокЭксплуатации,*

| *ОборудованиеВЭксплуатацииОстатки.СрокГодности КАК СрокГодности,*

«ЕСТЬNULL»(ОборудованиеВЭксплуатацииОстатки.КоличествоОстаток, 0) КАК

КоличествоОстаток

| **ИЗ**

| *ДанныеТЧ КАК ДанныеТЧ*

| **ЛЕВОЕ СОЕДИНЕНИЕ**

РегистрНакопления.ОборудованиеВЭксплуатации.Остатки(

| *& МоментВремени,*

| *Номенклатура В*

| **(ВЫБРАТЬ**

| *ДанныеТЧ.Номенклатура*

| **ИЗ**

| *ДанныеТЧ КАК ДанныеТЧ)*

| *И СрокЭксплуатации = ДАТАВРЕМЯ(1, 1, 1))*

КАК ОборудованиеВЭксплуатацииОстатки

| **ПО ДанныеТЧ.Номенклатура =**

ОборудованиеВЭксплуатацииОстатки.Номенклатура

| **УПОРЯДОЧИТЬ ПО**

| СрокГодности

| ИТОГИ

| МАКСИМУМ(Количество),

| СУММА(КоличествоОстаток)

| ПО

| Номенклатура";

Запрос.УстановитьПараметр("МоментВремени",?(Режим =
РежимПроведенияДокумента.Неоперативный, МоментВремени(), Неопределено));

Запрос.УстановитьПараметр("Ссылка", Ссылка);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаНоменклатура =

РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Пока ВыборкаНоменклатура.Следующий() Цикл

Нехватка = ВыборкаНоменклатура.Количество -

ВыборкаНоменклатура.КоличествоОстаток;

Если Нехватка > 0 Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Недостаточно оборудования " +

ВыборкаНоменклатура.Номенклатура + ". Не хватает " + Нехватка;

Сообщение.Сообщить();

Отказ = Истина;

КонецЕсли;

Если Отказ Тогда Продолжить

КонецЕсли;

ОсталосьСписать = ВыборкаНоменклатура.Количество;

ВыборкаДетальныеЗаписи = ВыборкаНоменклатура.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() И ОсталосьСписать > 0 Цикл

Если ВыборкаДетальныеЗаписи.СрокГодности < Дата Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "У оборудования " +

ВыборкаДетальныеЗаписи.Номенклатура + " срок годности истек " +

ВыборкаДетальныеЗаписи.СрокГодности;

Отказ = Истина;

КонецЕсли;

Если Отказ Тогда Продолжить

КонецЕсли;

КоличествоКСписанию = Мин(ОсталосьСписать,

ВыборкаДетальныеЗаписи.КоличествоОстаток);

// регистр ОборудованиеВЭксплуатации Расход

Движение = Движения.ОборудованиеВЭксплуатации.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.СрокГодности = ВыборкаДетальныеЗаписи.СрокГодности;

Движение.Количество = КоличествоКСписанию;

// регистр ОборудованиеВЭксплуатации Приход

Движение = Движения.ОборудованиеВЭксплуатации.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.СрокГодности = ВыборкаДетальныеЗаписи.СрокГодности;

Движение.СрокЭксплуатации = Дата +

*ВыборкаДетальныеЗаписи.СрокЭксплуатации * 24*3600;*

Движение.Количество = КоличествоКСписанию;

ОсталосьСписать = ОсталосьСписать - КоличествоКСписанию;

КонецЦикла;

КонецЦикла;

КонецПроцедуры

Прежде всего, нас интересует запрос (фрагмент выделен жирным шрифтом), рассмотрим его структуру и последовательность выполнения.

Первым шагом получим данные из табличной части документа. Для этого откроем консоль запросов и выберем поля «Номенклатура» и «Количество» из табличной части документа «Ввод в эксплуатацию». Для получения значения срока эксплуатации выберем поле «Номенклатура.СрокЭксплуатации (Рисунок 364).

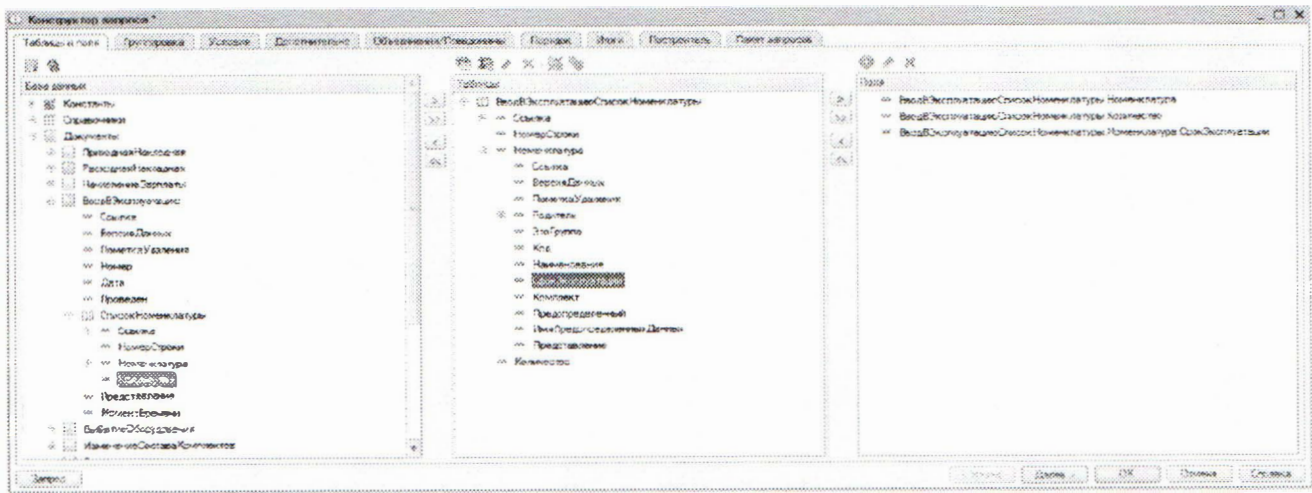


Рисунок 364

На вкладке «Условия» укажем ссылку на наш документ (Рисунок 365).

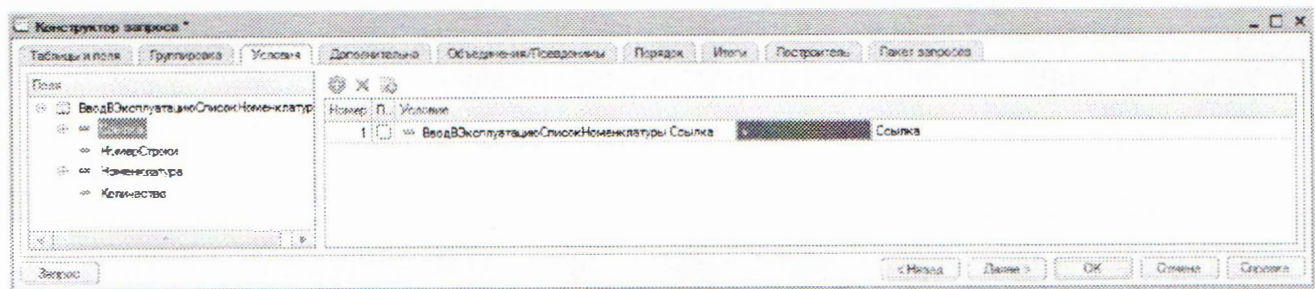


Рисунок 366

На случай наличия в базе нескольких одинаковых позиций, сгруппируем результат на вкладке «Группировка» по позициям (Рисунок 367):

- Номенклатура;
- Номенклатура.СрокЭксплуатации.

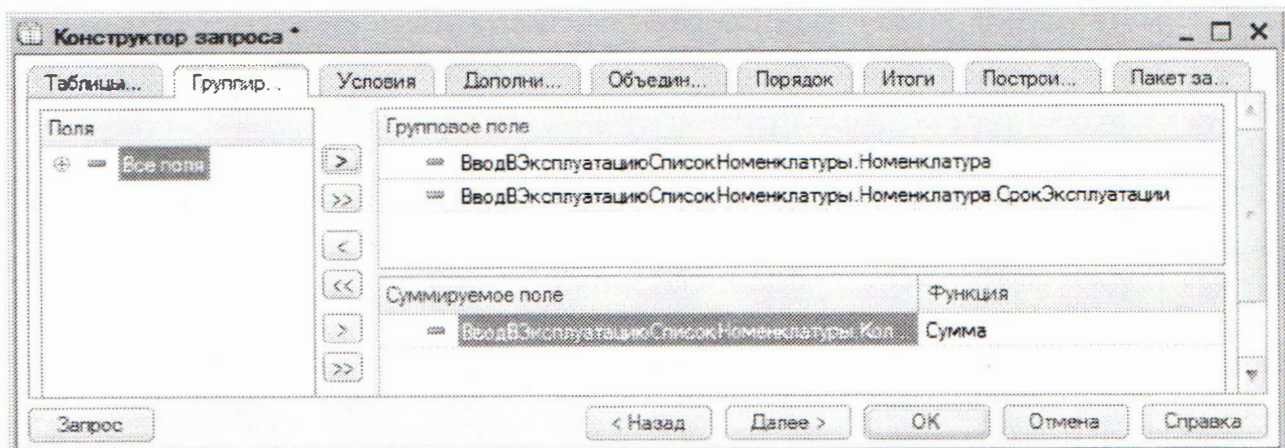


Рисунок 367

Поскольку в данном случае для каждой номенклатурной позиции срок годности одинаков, возможен еще один вариант группировки: сгруппировать только по номенклатуре, а в область «Суммируемое поле» добавить «СписокНоменклатуры.Номенклатура.СрокЭксплуатации» с применением функции «Максимум» (значения типа «Дата» просуммировать мы не можем). Так как для каждой

183 Верный старт в 1С www.work-1c.ru 88002344322 Бесплатный звонок

номенклатуры срок годности одинаков, то мы получим тот же результат, что и первом варианте группировки (Рисунок 368).

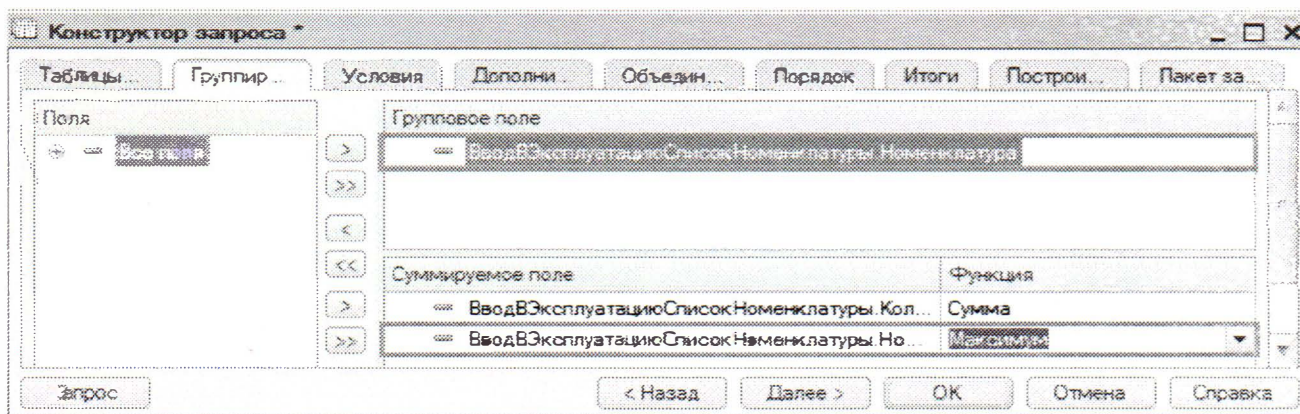


Рисунок 368

Перейдем на вкладку «Дополнительно» и создадим временную таблицу «ДанныеТЧ» (Рисунок 369).

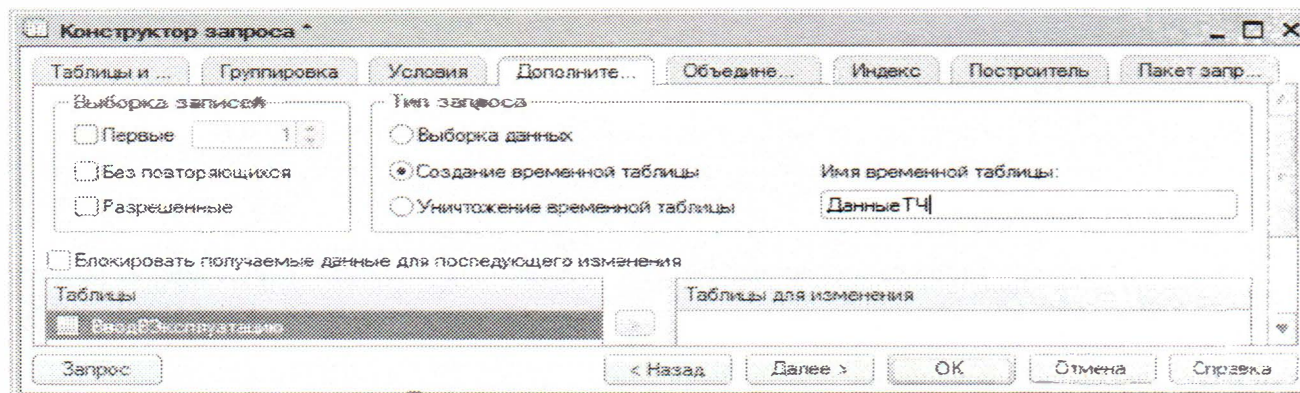


Рисунок 369

На вкладке «Пакет запроса» создадим новый пакет, на вкладке «Таблицы и поля» увидим временную таблицу «ДанныеТЧ» (Рисунок 370).

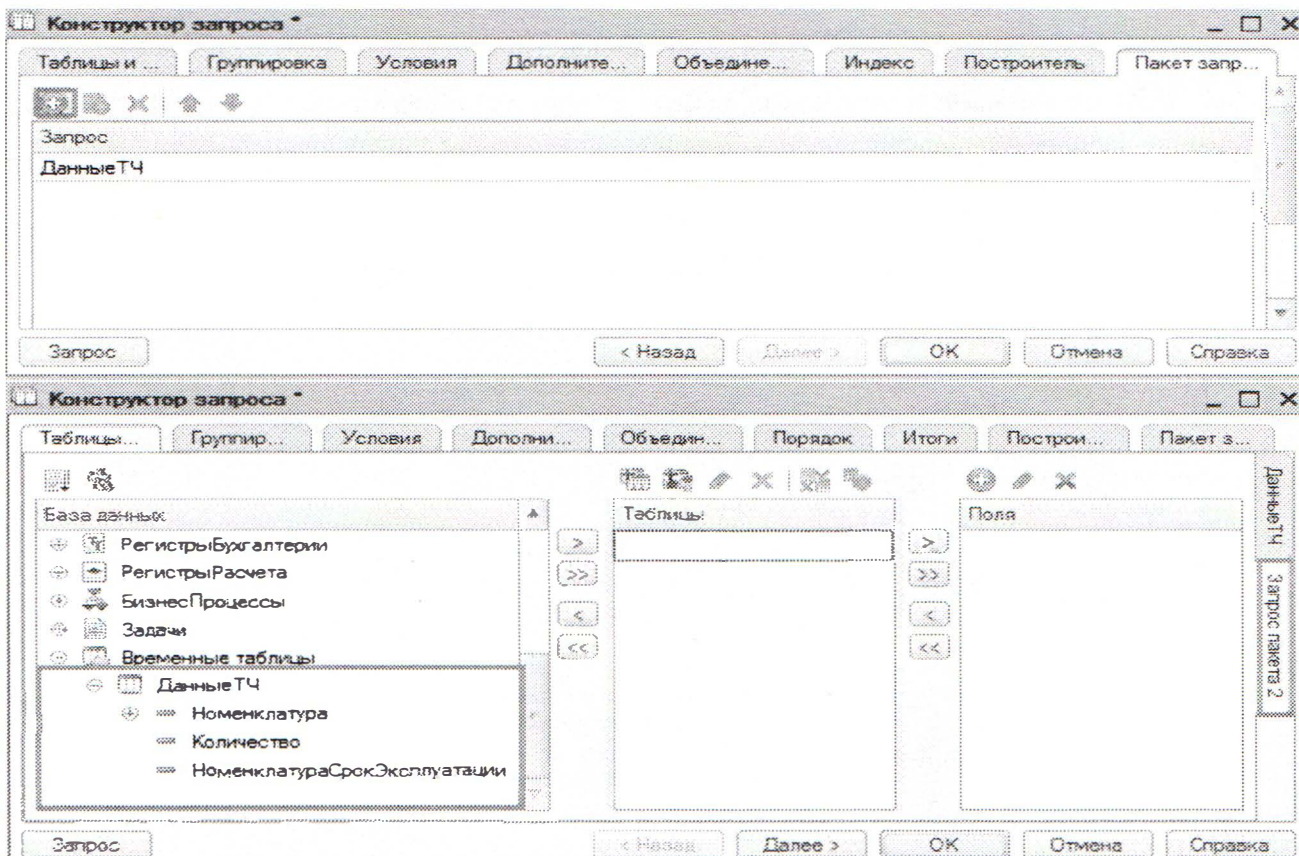


Рисунок 370

Таким образом, мы получили данные табличной части. Однако, оборудование, которое пришло по приходной накладной и у которого еще нет срока эксплуатации, поскольку в эксплуатацию оно еще не введено, хранится в регистре накопления «ОборудованиеВЭксплуатации». Обратимся к виртуальной таблице остатков данного регистра, так как нам нужны именно остатки на момент проведения документов.

В качестве параметров запроса укажем «Период» – момент времени, когда это движение происходило.

МоментВремени(), в отличие от даты, является более точной границей – границей на момент проведения документа, и состоит из даты и ссылки на документ (в случае если в базе содержится несколько документов на одну и ту же дату, по моменту времени можно четко идентифицировать необходимый нам документ) (Рисунок 371).

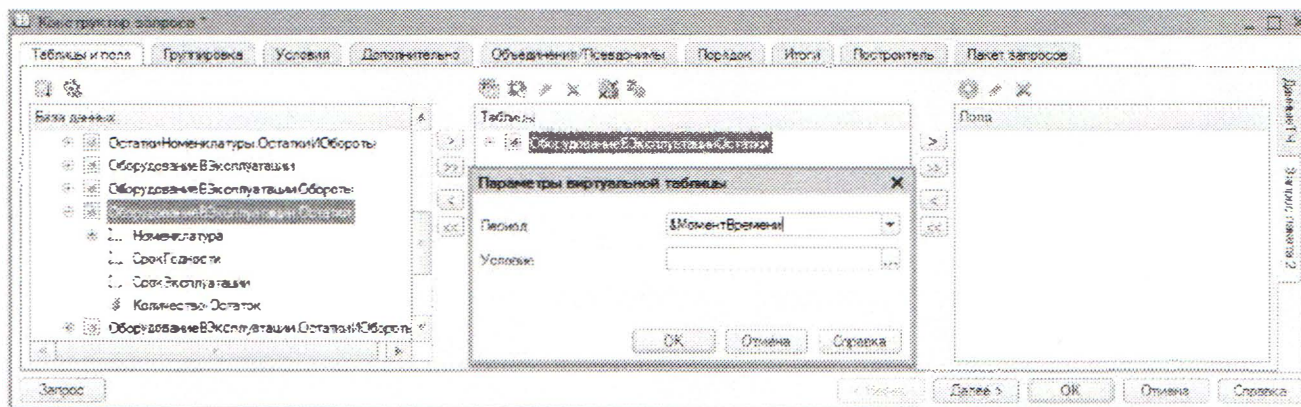


Рисунок 371

Далее ограничим получение данных только той номенклатурой, которая содержится в табличной части ввода в эксплуатацию. Для этого заполним параметр «Условие» данными из табличной части. Это можно сделать, вызвав форму заполнения условия (Рисунок 372).

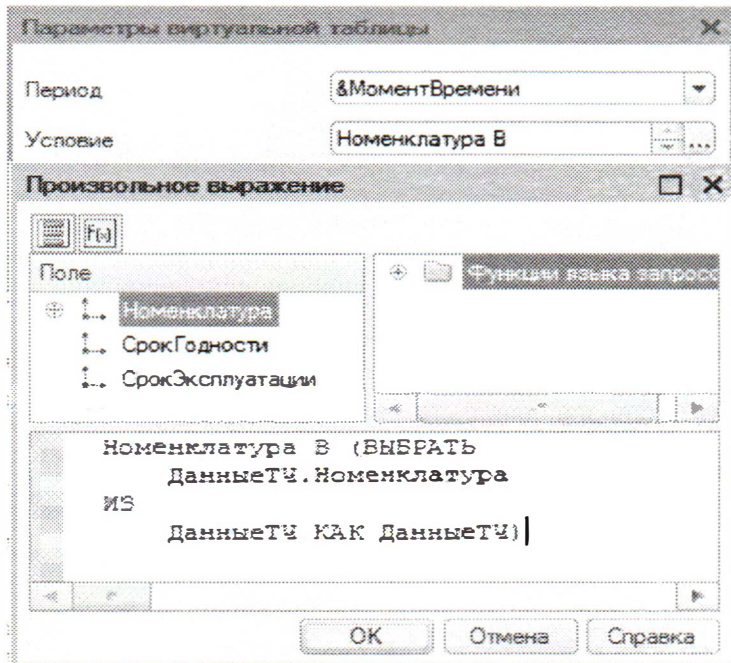


Рисунок 372

Рассмотрим другой случай работы с условиями.

Для этого создадим еще один пакет запросов и выберем поле «Номенклатура» из временной таблицы «Данные ТЧ» (Рисунок 373).

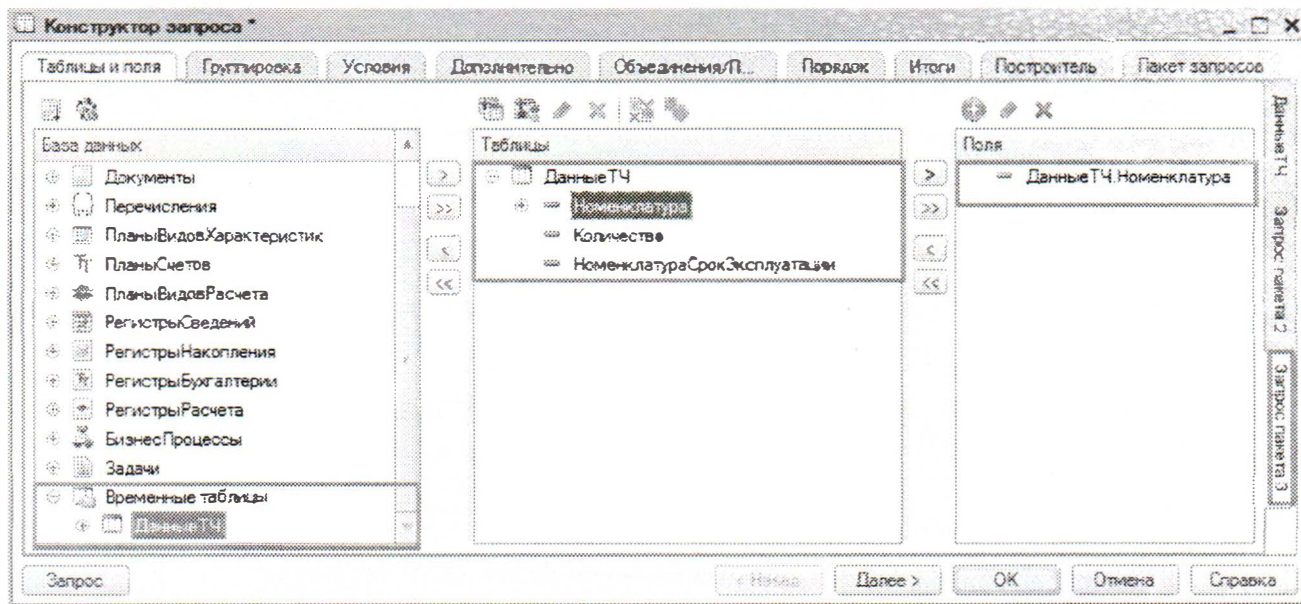


Рисунок 373

Скопируем этот фрагмент из текста запроса (Рисунок 374).

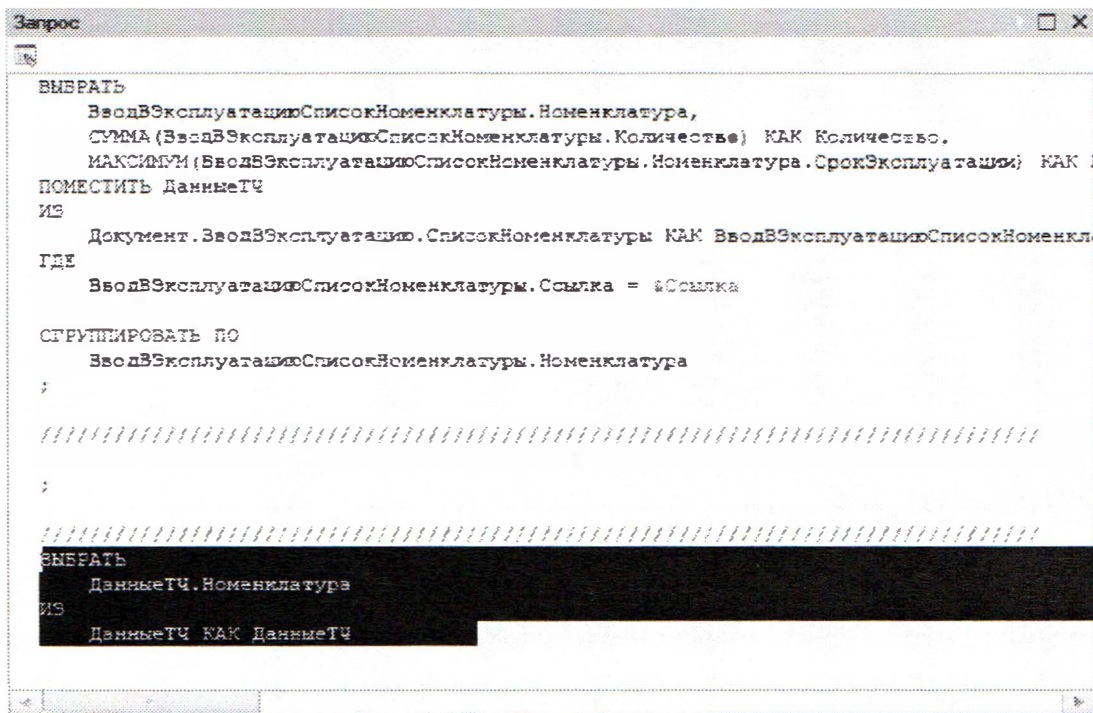


Рисунок 374

Скопируем данный фрагмент в поле описания условия запроса, добавив перед этим «Номенклатура В» и заключив в скобки текст запроса. Далее удалим пакет запроса 3. В результате получим то же условие, что и в первом варианте:

```
Номенклатура В (ВЫБРАТЬ
    ДанныеТЧ.Номенклатура
ИЗ
    ДанныеТЧ КАК ДанныеТЧ)
```

Обращаем ваше внимание, что в момент оприходования оборудования в документе «Приходная накладная» задается срок годности и, после проведения документа, в регистре ОборудованиеВЭксплуатацииОстатки созданной записи заполнено измерение «СрокГодности», но не заполнено измерение «СрокЭксплуатации». Это говорит о том, что оборудование еще не было введено в эксплуатацию. И, наоборот, если в записи регистра ОборудованиеВЭксплуатацииОстатки заполнено измерение «СрокЭксплуатации» – это значит, что оборудование уже было введено в эксплуатацию. В связи с этим, нам нужны только те записи регистра, по которым значение срока эксплуатации пока нет. Для этого дополним условие выражением через оператор «И»: *СрокЭксплуатации = ДатаВремя(1,1,1)*. Таким образом мы проверили срок годности на пустую дату (Рисунок 375).

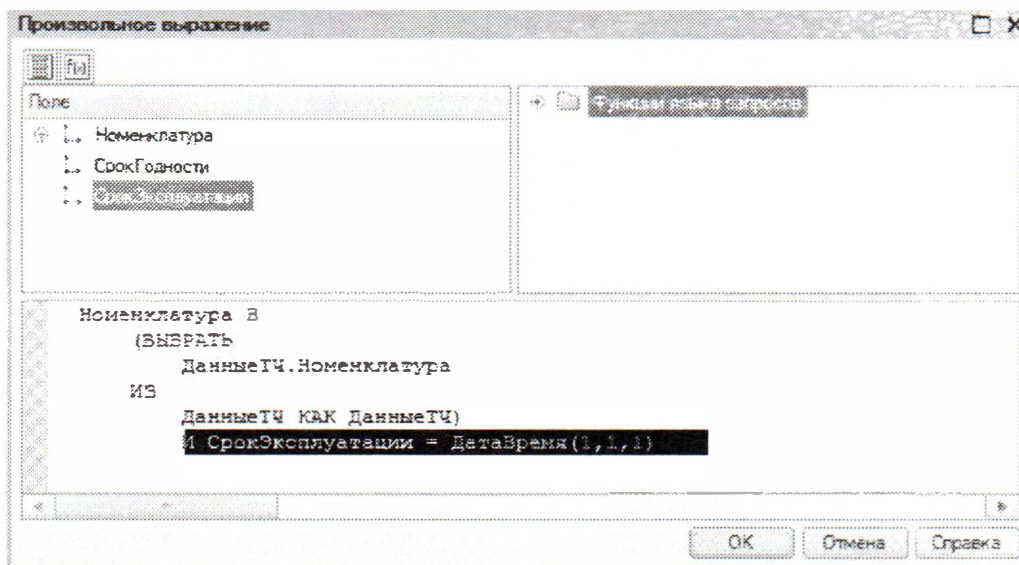


Рисунок 375

Смысл данного условия в том, что мы отобрали из всех записей только те, в которых на момент проведения документа содержится только номенклатура из табличной части, а срок эксплуатации является пустым значением.

Далее свяжем соединением по номенклатуре таблицы «Данные ТЧ» и таблицы остатков регистра «ОборудованиеВЭксплуатации». Для этого перетянем временную таблицу в поле таблиц запроса пакета 2, и получим итоговый запрос. Так как все данные, полученные из таблицы «Данные ТЧ», должны фигурировать в итоговом запросе, в отличие от полей таблицы «Оборудование в эксплуатации» (из этой таблицы нам интересны только те данные, которые содержатся в «Данные ТЧ»), используем левое соединение. В случае, если номенклатуры из таблицы «Оборудование в эксплуатации» нет на остатках, система выдаст сообщение пользователю. Соответственно, если мы свяжем таблицы не левым соединением, а внутренним, то система не выдаст сообщение пользователю при получении данных, которые есть в таблице «Данные ТЧ», но, при этом, отсутствуют в таблице «Оборудование в эксплуатации», так как у нас не будет записей по номенклатурным позициям, которых нет на остатках (Рисунок 376).

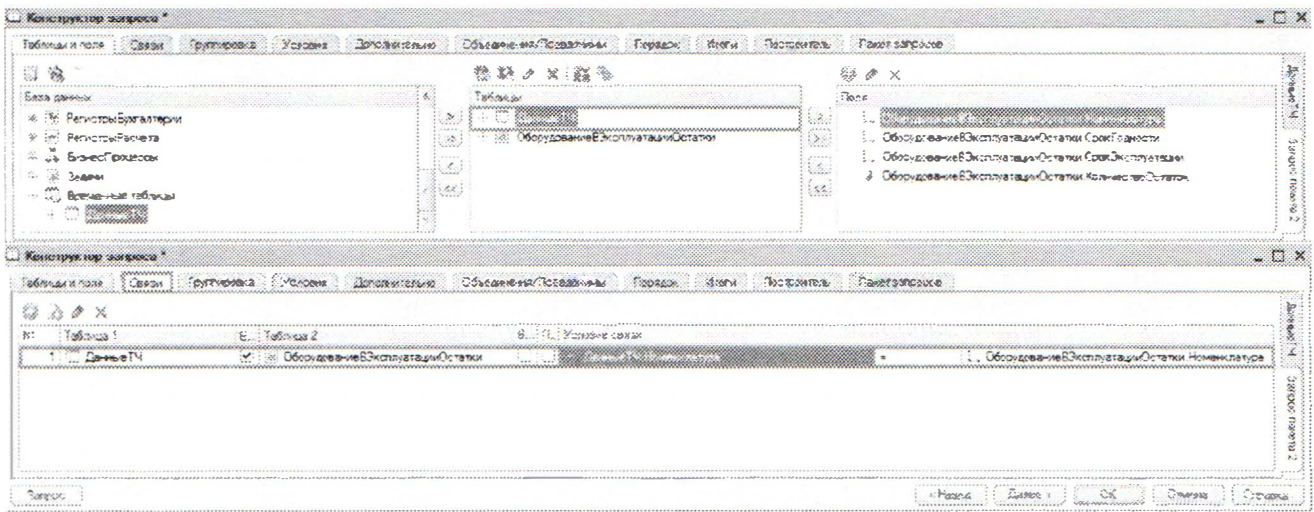


Рисунок 376

Определим состав полей, которые нужно получить в Запросе пакета 2.

Из таблицы «Данные ТЧ» нам понадобятся все поля (поля таблицы можно перетащить из области «База данных» конструктора запроса в область «Поля» без применения функции «ЕСТЬNULL», так как ранее мы связали таблицы левым соединением) (Рисунок 377).

Поля таблицы «Данные ТЧ»:

- ДанныеТЧ.Номенклатура;
- ДанныеТЧ.НоменклатураСрокЭксплуатации;
- ДанныеТЧ.НоменклатураСрокЭксплуатации (предварительно переименовав на вкладке «Объединения/Псевдонимы» в «ДанныеТЧ.СрокЭксплуатации», для более легкого восприятия).

Из таблицы «ОборудованиеВЭксплуатацииОстатки» нам понадобятся:

- ОборудованиеВЭксплуатацииОстатки.СрокГодности (с этим сроком годности будем списывать и оприходовать номенклатуру, но уже со сроком эксплуатации);
- ОборудованиеВЭксплуатацииОстатки.КоличествоОстаток.

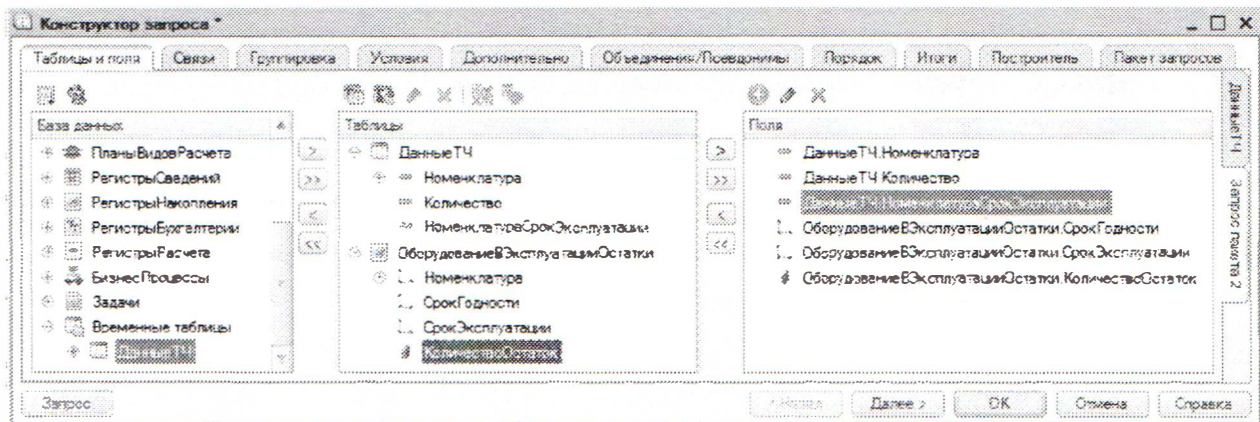


Рисунок 377

Как видим, на закладке «Объединения/псевдонимы» у нас прописаны псевдонимы для всех полей (Рисунок 378).

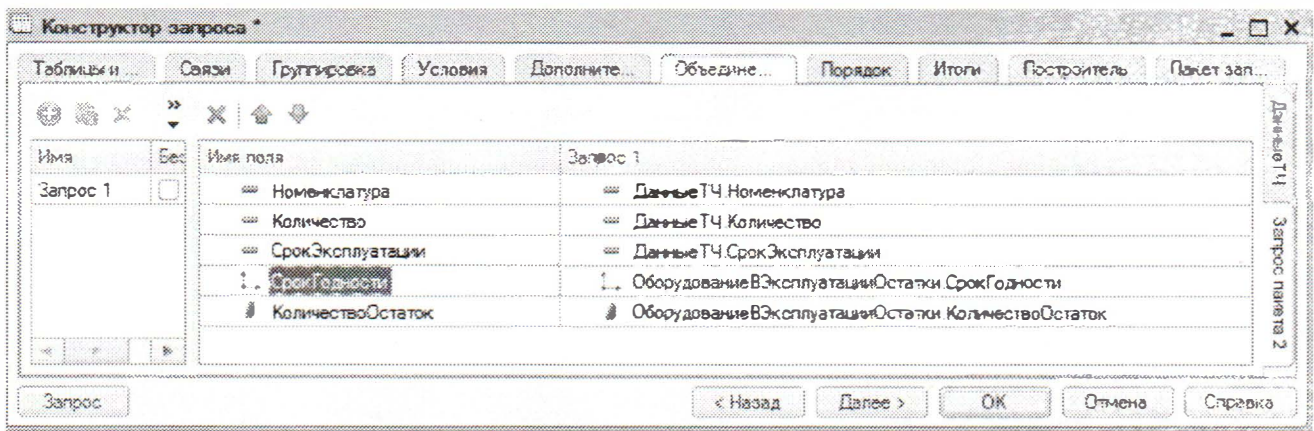


Рисунок 378

Обратим внимание, что если мы что-то изменим в одном из полей, то имя поля изменится на «Поле1» (Рисунок 379).

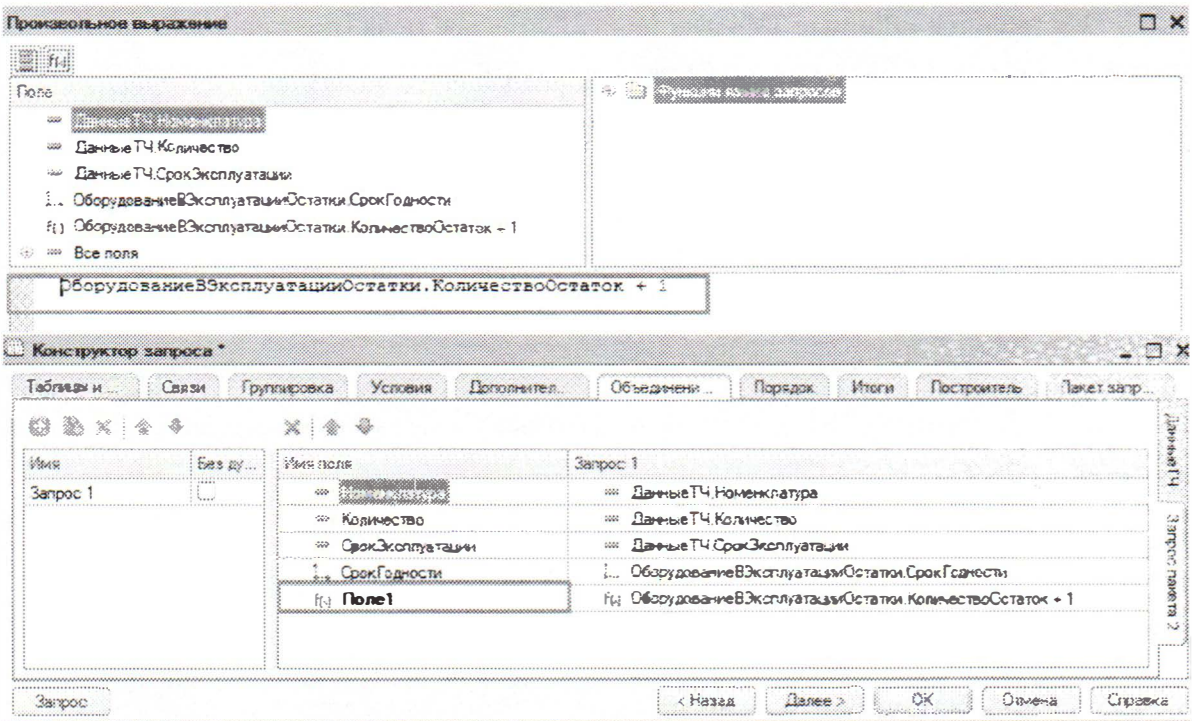


Рисунок 379

Данное наименование поля не очень удобно для использования в работе, поэтому его можно переименовать вручную на закладке «Объединения/псевдонимы», после этого оно меняться не будет. При изменении значения поля либо его переименовании наименование выделяется жирным шрифтом. Это означает, что в тексте запроса появился данный псевдоним через оператор «КАК» (Рисунок 380).

```

Документ ВводВЭксплуатацию. Модуль объекта
ВводВЭксплуатацию. Заблокировать () :
Запрос = Новый Запрос;
Запрос.Текст =
*ВЫБРАТЬ
: ВводВЭксплуатациюСписокНоменклатуры.Номенклатура КАК Номенклатура,
: СУММА (ВводВЭксплуатациюСписокНоменклатуры.Количество) КАК Количество,
: ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации КАК СрокЭксплуатации
: ПОНЕСИТЬ ДАННЫЕ ТЧ
: ИЗ
: Документ.ВводВЭксплуатацию.СписокНоменклатуры КАК ВводВЭксплуатациюСписокНоменклатуры
: ГДЕ
: ВводВЭксплуатациюСписокНоменклатуры.Ссылка = &Ссылка

: СГРУППИРОВАТЬ ПО
: ВводВЭксплуатациюСписокНоменклатуры.Номенклатура,
: ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации

: ИНДЕКСИРОВАТЬ ПО
: Номенклатура
: ;

////////////////////////////////////
: ВЫБРАТЬ
: ДанныеТЧ.Номенклатура,
: ДанныеТЧ.Количество,
: ДанныеТЧ.СрокЭксплуатации,
: ОборудованиеВЭксплуатацииОстатки.СрокГодности,
: ОборудованиеВЭксплуатацииОстатки.КоличествоОстаток + 1 КАК После
: ИЗ
: ДанныеТЧ КАК ДанныеТЧ
: ЛЕВСЕ СОЕДИНЕНИЕ РегистрНакопления.ОборудованиеВЭксплуатации.Остатки (
: «МоментВремени,
: Номенклатура В
: (ВЫБРАТЬ
: ДанныеТЧ.Номенклатура
: ИЗ
: ДанныеТЧ КАК ДанныеТЧ)
: И СрокЭксплуатации = ДАТАВРЕМЯ(1, 1, 1)) КАК ОборудованиеВЭксплуатацииОстатки
: ПО ДанныеТЧ.Номенклатура = ОборудованиеВЭксплуатацииОстатки.Номенклатура"

```

Рисунок 380

Теперь следует подумать над значением поля «ОборудованиеВЭксплуатацииОстатки.КоличествоОстаток». Согласно логике задачи, в момент проведения документа система должна выполнять проверку на наличие остатка на складе, соответственно, будет применена операция сравнения. Следовательно, значение «NULL» для данного поля будет некорректным и приведет к ошибкам, во избежание которых используем «ЕСТЬNULL» (если в таблице «Данные ТЧ» значение полей равным «NULL» быть не может, то в таблице остатков «ОборудованиеВЭксплуатации» вполне могут оказаться нулевые остатки и эта ситуация требует обработки) (Рисунок 381).

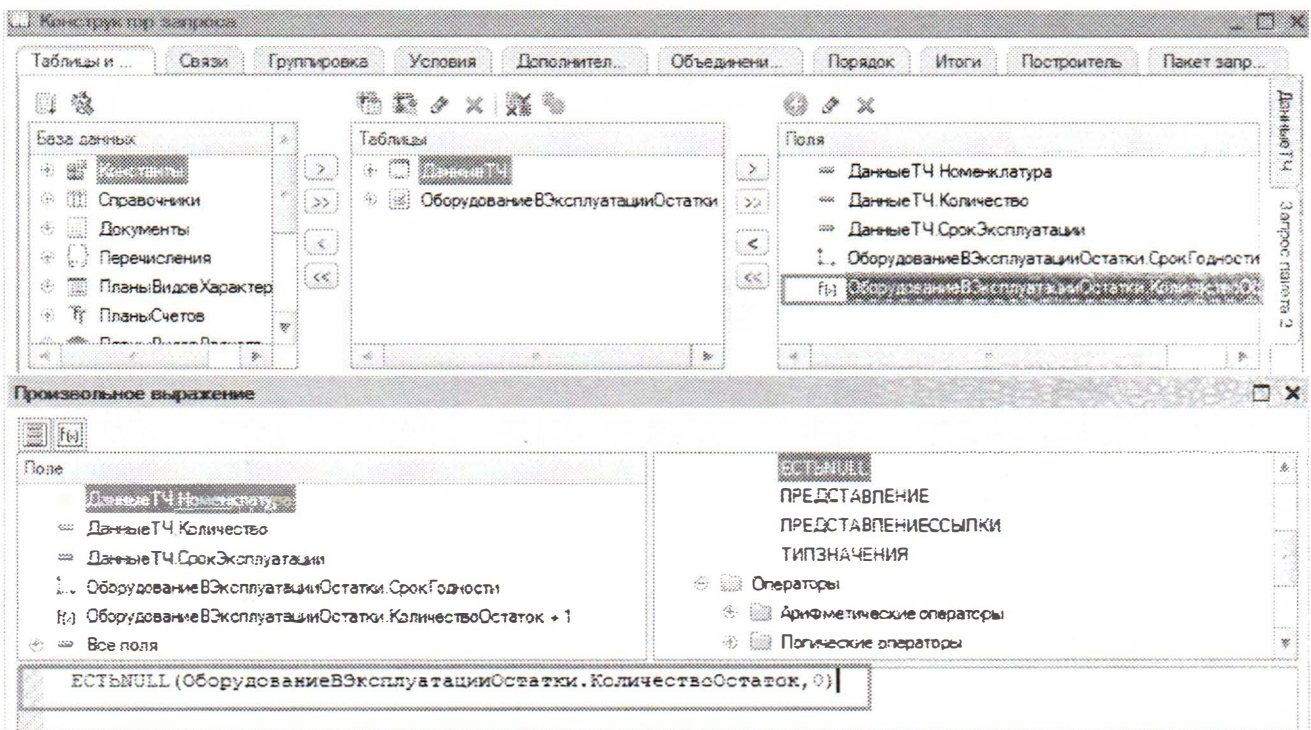


Рисунок 381

Таким образом, если в момент проведения документа в базе будут нулевые остатки, система выведет об этом сообщение, и далее с этим документом мы работать не будем.

По такой же логике, поле «ОборудованиеВЭксплуатацииОстатки.СрокГодности» теоретически тоже может быть равным «NULL», но, в данном случае, использовать «ЕСТЬNULL» необязательно, поскольку если срока годности не будет и количества недостаточно на остатках, дальнейшая работа с документом прекращается. Однако, применение «ЕСТЬNULL» к сроку годности ошибкой не является и, если в работе требуется унифицировать код, во избежание ошибок можно применять к аналогичным случаям «ЕСТЬNULL».

Одно из условий задачи – в первую очередь вводить в эксплуатацию оборудование с минимальным сроком годности. Для его реализации на закладке «Порядок» конструктора запросов требуется упорядочить записи поля «Срок годности» по возрастанию (поскольку в измерение «Срок годности» регистра «ОборудованиеВЭксплуатации» записывается дата истечения срока годности, требуется упорядочивать по возрастанию) (Рисунок 382).

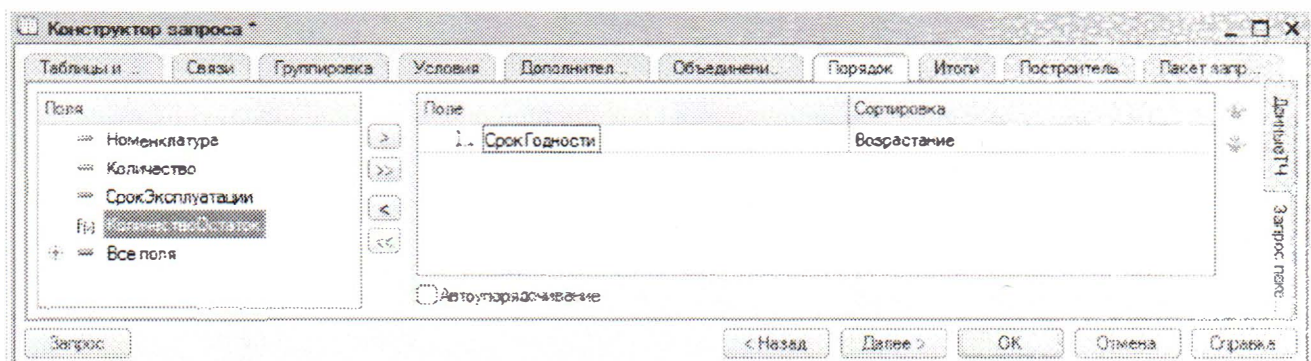


Рисунок 382

Обратим внимание, что одной номенклатуре может соответствовать несколько записей с различными сроками годности и сроками эксплуатации, но, тем не менее, когда мы будем проверять, достаточен ли остаток на складе для проведения документа, нам потребуются общие данные по номенклатуре из таблицы «Данные ТЧ». Следовательно, нам также нужны и итоги.

Для настройки итогов на закладке «Итоги» конструктора запросов требуется в качестве группировочного поля выбрать номенклатуру, а в качестве итогов – указать, какое количество есть в документе, и какое количество есть на остатках (Рисунок 383).

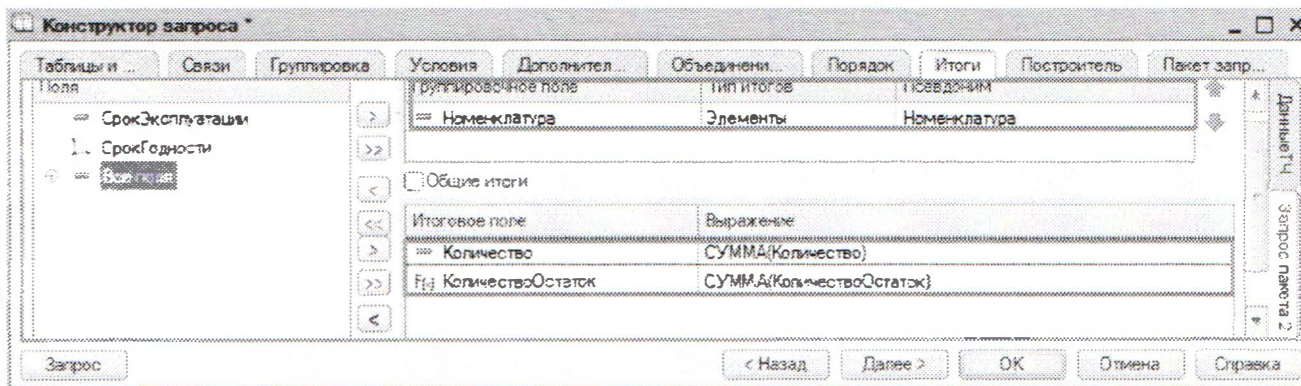


Рисунок 383

При этом, когда мы связываем таблицы «Данные ТЧ» и «Оборудование в эксплуатации», количество номенклатуры, которое есть в документе, может задублироваться, поскольку в таблице «Оборудование в эксплуатации» на одну номенклатуру может быть множество строк, и позиции из «Данные ТЧ», по этой причине, могут повторяться. Следовательно, мы не можем использовать функцию СУММА применительно к полю «Количество», рекомендуется использовать функцию МАКСИМУМ для данных табличной части (Рисунок 384).

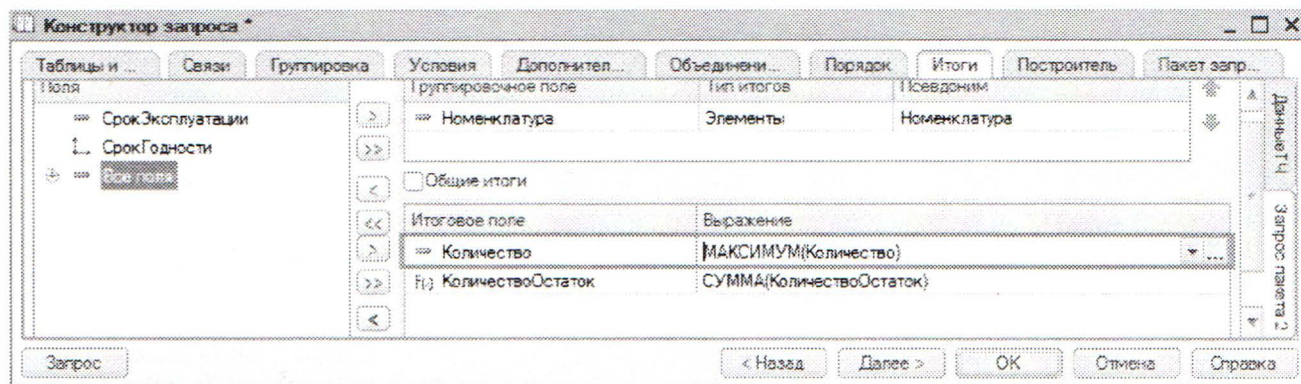


Рисунок 384

В результате получим итоговый текст запроса:

"ВЫБРАТЬ

| ВводВЭксплуатациюСписокНоменклатуры.Номенклатура КАК Номенклатура,
| СУММА(ВводВЭксплуатациюСписокНоменклатуры.Количество) КАК Количество,

| МАКСИМУМ(Количество),
| СУММА(КоличествоОстаток)
| ПО
| Номенклатура"

Посмотрим на дальнейшую обработку результата запроса:

Пока ВыборкаНоменклатура.Следующий() Цикл

Нехватка = ВыборкаНоменклатура.Количество -

ВыборкаНоменклатура.КоличествоОстаток;

Если Нехватка > 0 Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Недостаточно оборудования " +

ВыборкаНоменклатура.Номенклатура + ". Не хватает " + Нехватка;

Сообщение.Сообщить();

Отказ = Истина;

КонецЕсли;

Если Отказ Тогда Продолжить

КонецЕсли;

ОсталосьСписать = ВыборкаНоменклатура.Количество;

ВыборкаДетальныеЗаписи = ВыборкаНоменклатура.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() И ОсталосьСписать > 0 Цикл

Если ВыборкаДетальныеЗаписи.СрокГодности < Дата Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "У оборудования " +

ВыборкаДетальныеЗаписи.Номенклатура + " срок годности истек " + ВыборкаДетальныеЗаписи.СрокГодности;

Отказ = Истина;

КонецЕсли;

Если Отказ Тогда Продолжить

КонецЕсли;

КоличествоКСписанию = Мин(ОсталосьСписать,

ВыборкаДетальныеЗаписи.КоличествоОстаток);

// регистр ОборудованиеВЭксплуатации Расход

Движение = Движения.ОборудованиеВЭксплуатации.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.СрокГодности = ВыборкаДетальныеЗаписи.СрокГодности;

Движение.Количество = КоличествоКСписанию;

// регистр ОборудованиеВЭксплуатации Приход

Движение = Движения.ОборудованиеВЭксплуатации.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.СрокГодности = ВыборкаДетальныеЗаписи.СрокГодности;

Движение.СрокЭксплуатации = Дата +

ВыборкаДетальныеЗаписи.СрокЭксплуатации * 24*3600;

Движение.Количество = КоличествоКСписанию;

ОсталосьСписать = ОсталосьСписать - КоличествоКСписанию;

КонецЦикла;

КонецЦикла;

В обработке результата запроса мы проходим в цикле по детальным записям по номенклатуре, проверяем остатки в данном фрагменте кода (Рисунок 385).

```
Документ: ВводВЭксплуатацию: Модуль объекта
Запрос.УстановитьПараметр("Ссылка", Ссылка);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаКонтракты = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Илика ВыборкаКонтракты.Следующий(); Илика
Наквотка = ВыборкаКонтракты.Количество - ВыборкаКонтракты.КоличествоОстатки;
Если Наквотка > 0 Тогда
    Сообщение * Новой СообщенииПользователю;
    Сообщение.Текст = "Недостаточно оборудования " + ВыборкаКонтракты.Номенклатура + ". Не хватает " + Наквотка;
    Сообщением.Сообщить();
    Отказ = Илика;
КонецЕсли;

Если Отказ Тогда Прервать;
КонецЕсли;

ОсталосьСписать = ВыборкаКонтракты.Количество;
ВыборкаДетальныеЗаписи = ВыборкаКонтракты.Выбрать();

Илика ВыборкаДетальныеЗаписи.Следующий(); Илика
Если ВыборкаДетальныеЗаписи.СрокГодности < Дата Тогда
    Сообщение * Новой СообщенииПользователю;
    Сообщение.Текст = "У оборудования " + ВыборкаДетальныеЗаписи.Номенклатура + " срок годности истек " + ВыборкаДетальныеЗаписи.СрокГодности;
    Отказ = Илика;
КонецЕсли;

Если Отказ Тогда Прервать;
КонецЕсли;

КоличествоКСписанию = Илика(ОсталосьСписать, ВыборкаДетальныеЗаписи.КоличествоОстатки);

// регистр ОборудованиеВЭксплуатации Расход
Движение = Движения.ОборудованиеВЭксплуатации.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.СрокГодности = ВыборкаДетальныеЗаписи.СрокГодности;
Движение.Количество = КоличествоКСписанию;
```

Рисунок 385

5.1.2 Индексирование.

Рассмотрим индексацию, которая имеет смысл только для тех полей, по которым связаны таблицы запроса. Поскольку таблицы «Данные ТЧ» и

«ОборудованиеВЭксплуатацииОстатки» в рамках запроса пакета 2 связаны по полю «Номенклатура», индексировать мы также будем по данному полю (Рисунок 386).

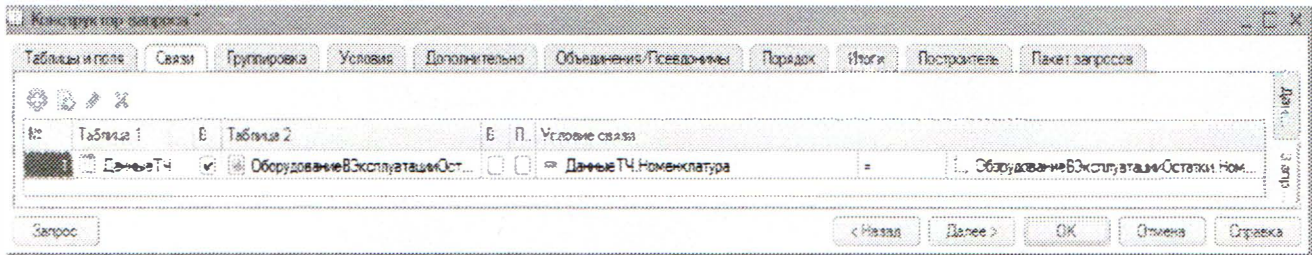


Рисунок 386

Для небольших таблиц большого смысла данное действие не имеет, однако, для таблиц с большим количеством записей индексация может значительно ускорить время исполнения запроса.

Перейдем на закладку «Индекс» конструктора запросов и добавим поле «Номенклатура» (Рисунок 387).

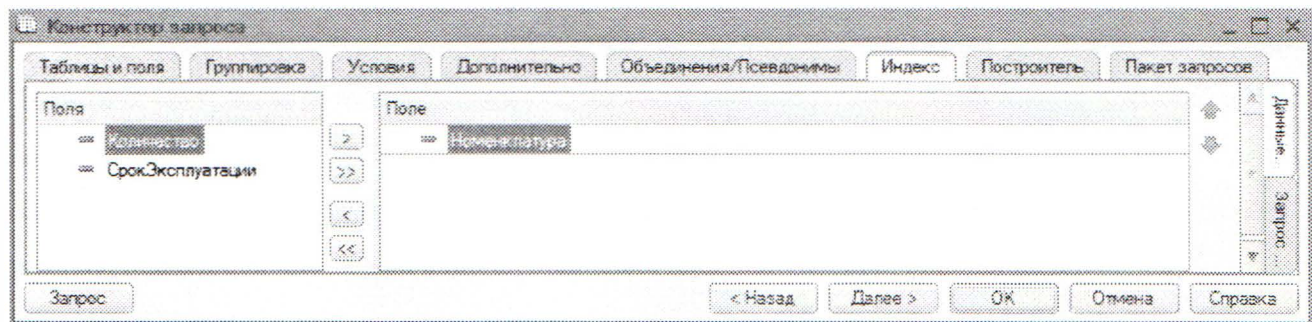


Рисунок 387

Итоговый текст запроса выглядит следующим образом:

"ВЫБРАТЬ

| ВводВЭксплуатациюСписокНоменклатуры.Номенклатура КАК Номенклатура,
 | СУММА(ВводВЭксплуатациюСписокНоменклатуры.Количество) КАК Количество,
 | ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации КАК

СрокЭксплуатации

| ПОМЕСТИТЬ ДанныеТЧ

| ИЗ

| Документ.ВводВЭксплуатацию.СписокНоменклатуры КАК

ВводВЭксплуатациюСписокНоменклатуры

| ГДЕ

| ВводВЭксплуатациюСписокНоменклатуры.Ссылка = &Ссылка

| СГРУППИРОВАТЬ ПО

| ВводВЭксплуатациюСписокНоменклатуры.Номенклатура,

| ВводВЭксплуатациюСписокНоменклатуры.Номенклатура.СрокЭксплуатации

| ИНДЕКСИРОВАТЬ ПО

Номенклатура

;

////////////////////////////////////

ВЫБРАТЬ

ДанныеТЧ.Номенклатура КАК Номенклатура,

ДанныеТЧ.Количество КАК Количество,

ДанныеТЧ.СрокЭксплуатации,

ОборудованиеВЭксплуатацииОстатки.СрокГодности КАК СрокГодности.

«ЕСТЬNULL.»(ОборудованиеВЭксплуатацииОстатки.КоличествоОстаток, 0) КАК

КоличествоОстаток

ИЗ

ДанныеТЧ КАК ДанныеТЧ

ЛЕВОЕ СОЕДИНЕНИЕ

РегистрНакопления.ОборудованиеВЭксплуатации.Остатки(

&МоментВремени,

Номенклатура В

(ВЫБРАТЬ

ДанныеТЧ.Номенклатура

ИЗ

ДанныеТЧ КАК ДанныеТЧ)

И СрокЭксплуатации = ДАТАВРЕМЯ(1, 1, 1)) КАК

ОборудованиеВЭксплуатацииОстатки

ПО ДанныеТЧ.Номенклатура =

ОборудованиеВЭксплуатацииОстатки.Номенклатура

УПОРЯДОЧИТЬ ПО

СрокГодности

ИТОГИ

МАКСИМУМ(Количество),

СУММА(КоличествоОстаток)

ПО

Номенклатура";

5.1.3 Выбытие оборудования.

Далее разберем запрос из документа «Выбытие оборудования». Для этого откроем модуль объекта документа и перейдем в обработку проведения (Рисунок 388).

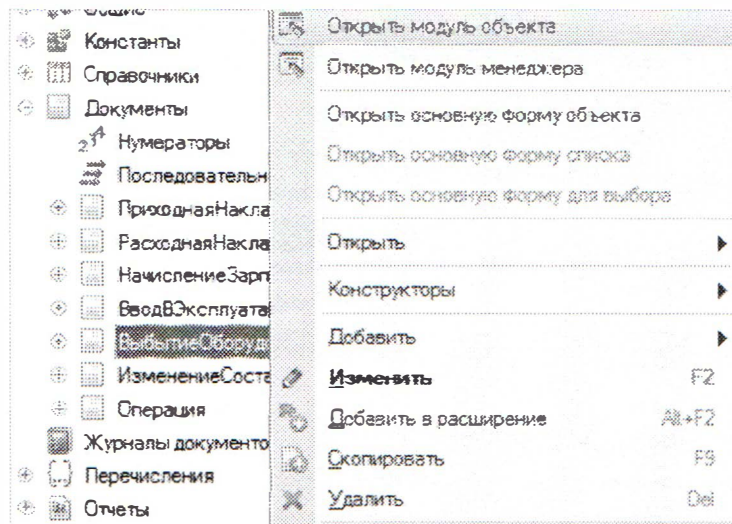


Рисунок 388

Удалим запрос из процедуры и вызовем новый конструктор запроса.

Следует обратить внимание на следующие нюансы задачи: сложность в том, что в задаче прописано условие расчета себестоимости по средней. Следовательно, при проектировании структуры метаданных, мы не можем использовать для хранения данных расчета себестоимости регистр «Оборудование в эксплуатации», поскольку в нем содержатся такие измерения как: «срок годности» и «срок эксплуатации». Если мы поместим данные по себестоимости в этот регистр, то она будет рассчитываться в разрезе данных показателей, а не по средней. В связи с этим, в структуре метаданных конфигурации содержится еще один регистр «ОстаткиНоменклатуры» (Рисунок 389).

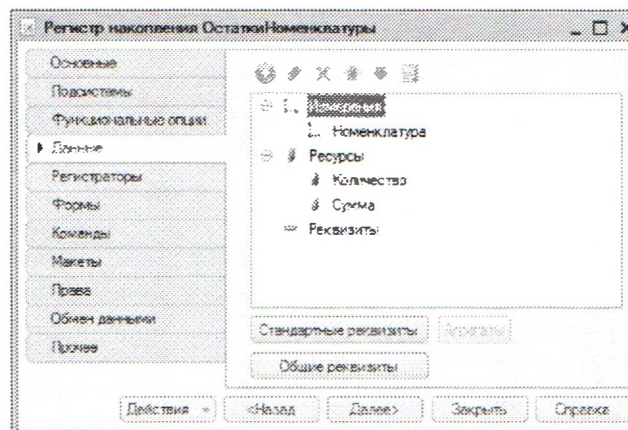


Рисунок 389

Соответственно, когда мы приходим документ по «Приходной накладной», документ делает движения по 2-м регистрам (Рисунок 390).

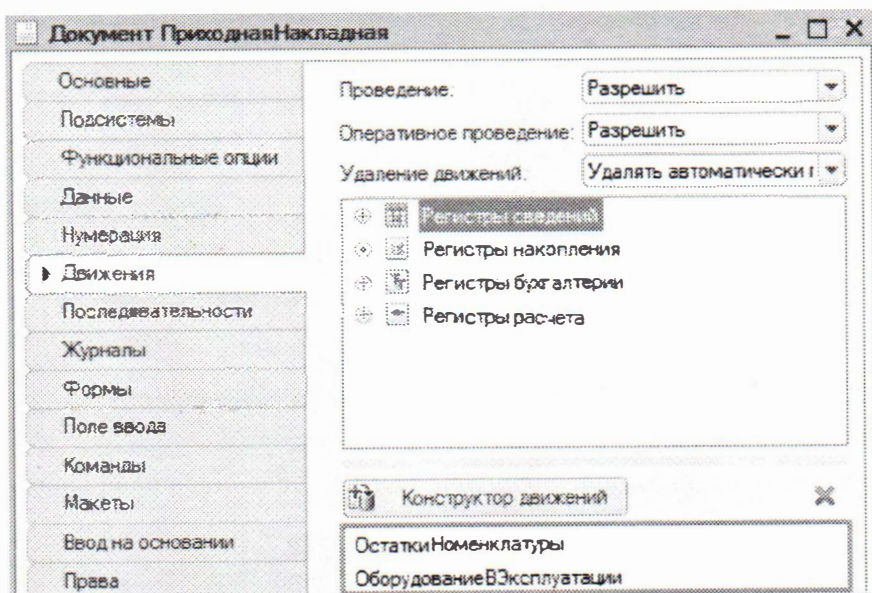


Рисунок 390

В связи с этим, в обработке проведения документа «Выбытие номенклатуры» нам нужно получать данные из таблицы «Оборудование в эксплуатации», в которой нас интересуют данные по тем номенклатурным позициям, по которым просрочены срок годности или срок эксплуатации; из таблицы «Остатки номенклатуры» – данные по сумме оборудования.

Перейдем к построению запроса.

Для этого в конструкторе запроса обратимся к таблице остатков регистра «Оборудование ВЭксплуатации» и зададим параметры виртуальной таблицы (Рисунок 391).

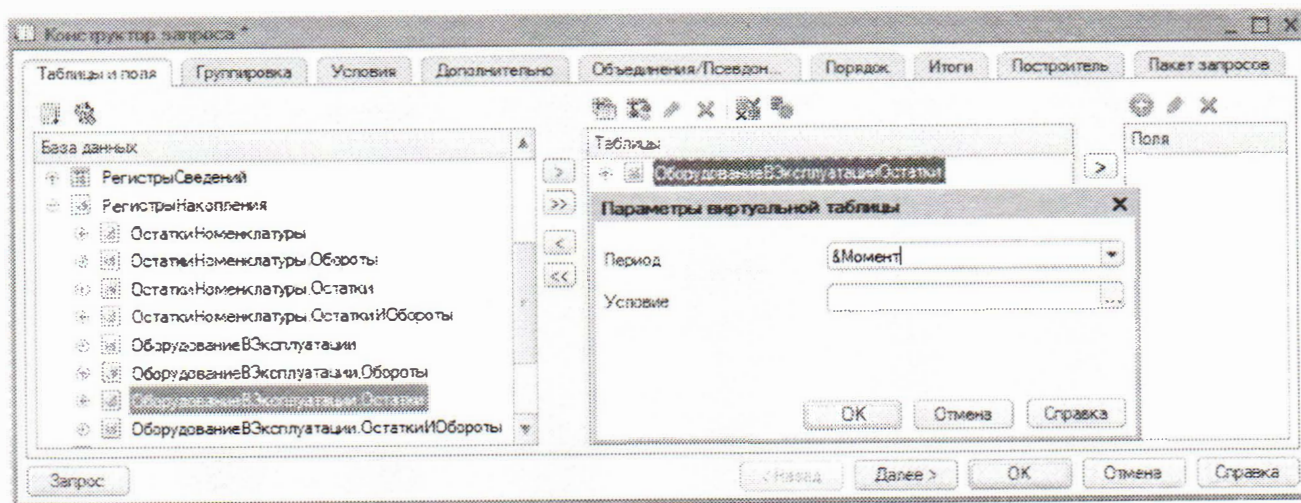


Рисунок 391

Важно обратить внимание на условие, которое нужно накладывать на срок годности и срок эксплуатации, которые должны будут истечь до момента, заданного в параметрах виртуальной таблицы (Рисунок 392).

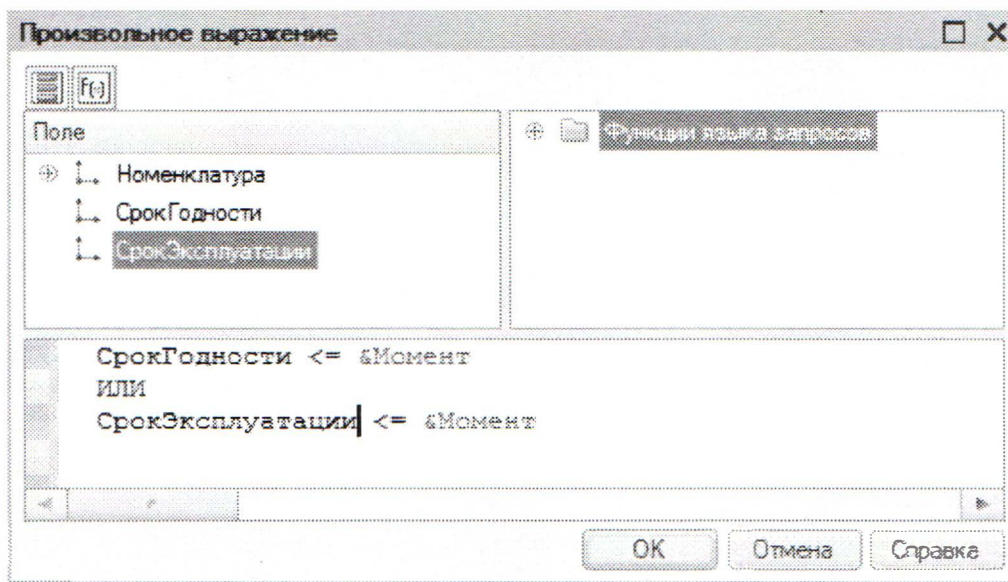


Рисунок 392

Здесь следует обратить внимание на следующее: у оборудования, которое еще не введено в эксплуатацию, значение реквизита «СрокЭксплуатации» не задан и равен пустому значению. Как мы помним, применительно к значению типа «Дата», пустое значение – это 1 января 0001 года, что раньше любой даты. Это значит, что таким условием мы спишем все оборудование, для которого срок эксплуатации не задан, то есть все оборудование, которое зарегистрировано в базе по приходной накладной, но еще не введено в эксплуатацию.

Это условие не удовлетворяет решению задачи, поэтому требуется изменить вторую часть условия после оператора «ИЛИ». Часть условия по сроку годности удовлетворяет решению задачи, так как срок годности у оборудования остается неизменным и его истечение не зависит от того, находится оборудование в эксплуатации или нет.

Усложним выражение по сроку эксплуатации, сравнив его с пустой датой (Рисунок 393).

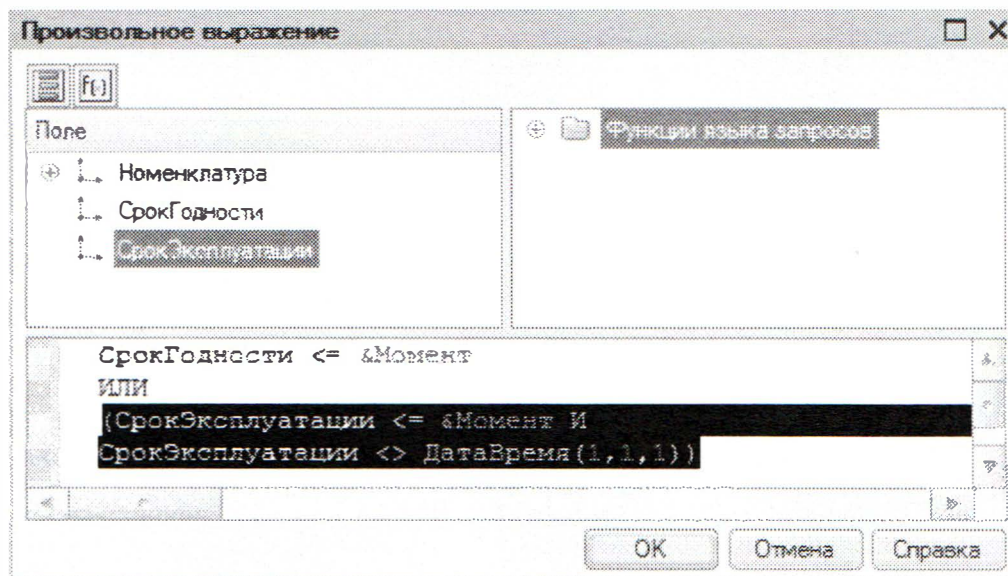


Рисунок 393

Далее выберем все поля из таблицы «ОборудованиеВЭксплуатации» и поместим их во временную таблицу «Оборудование» (Рисунок 394).

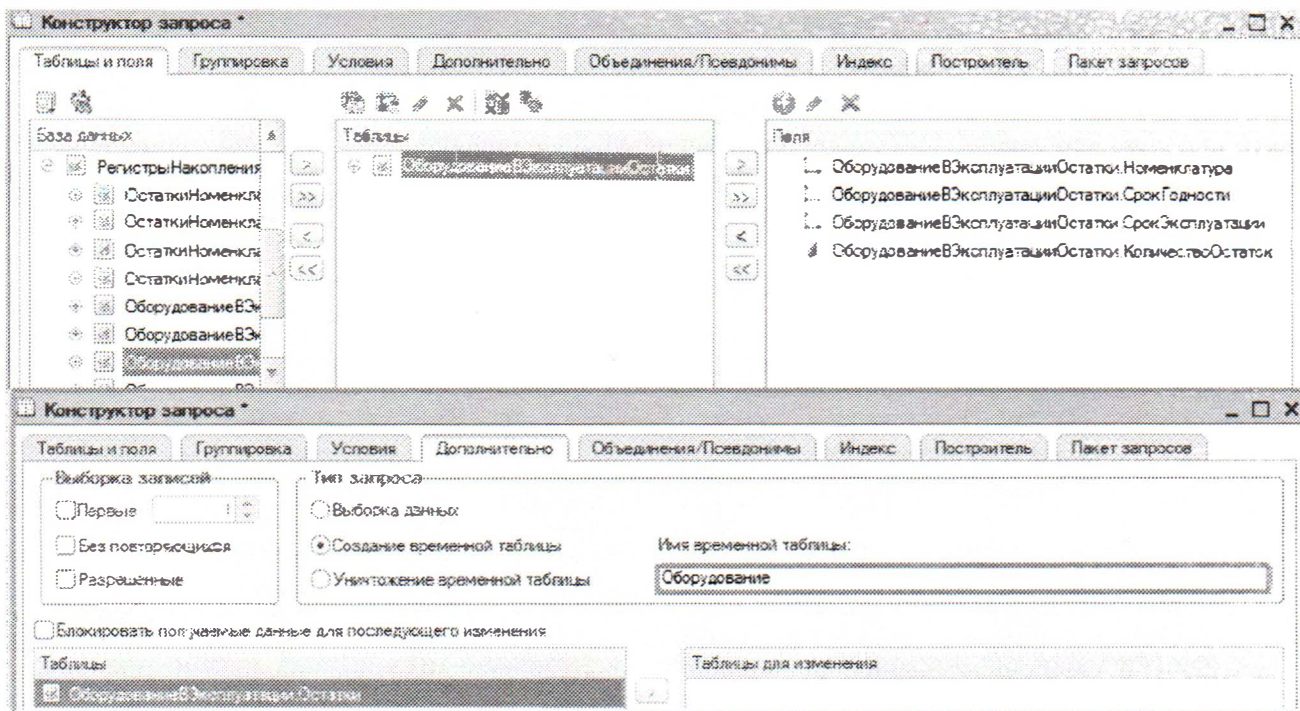


Рисунок 394

Сразу проиндексируем поля, по которым планируем связывать таблицы и добавим новый пакет запросов (Рисунок 395-396).

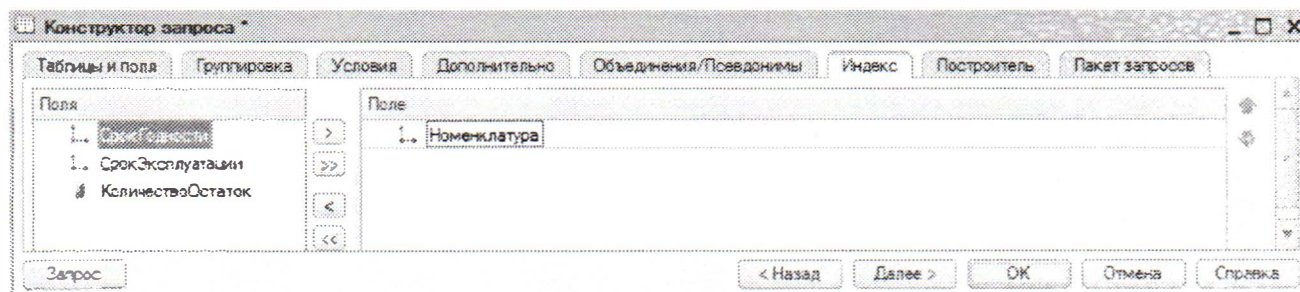


Рисунок 395

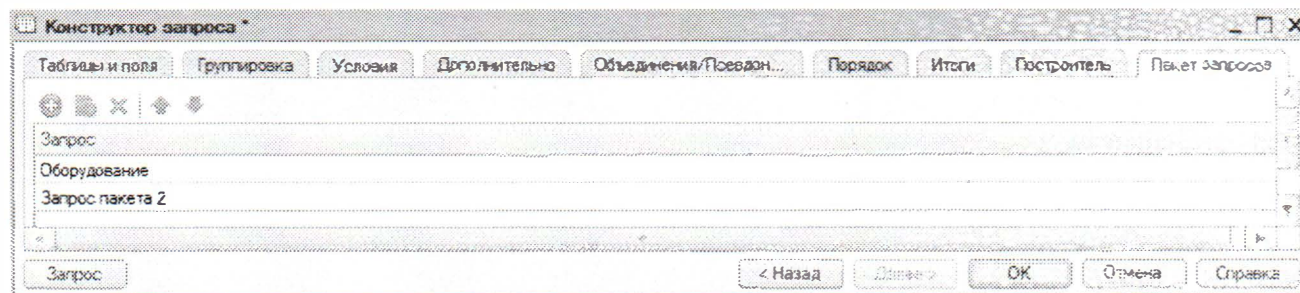


Рисунок 396

Определим состав таблиц пакетного Запроса пакета2. Нас интересует виртуальная таблица остатков регистра «ОстаткиНоменклатуры», в которой зададим тот же период, что и в таблице «Оборудование» (Рисунок 397).

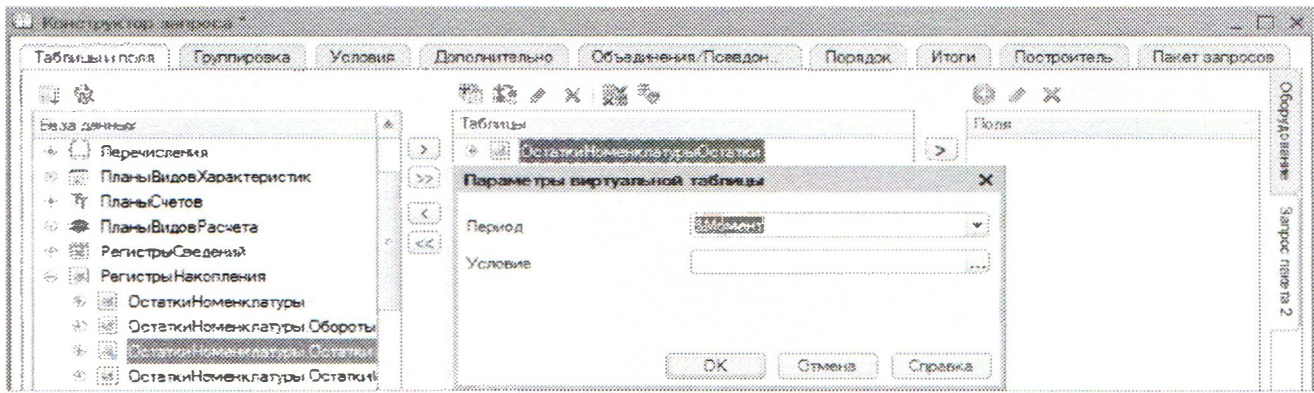


Рисунок 397

Далее создадим еще один запрос пакета, получим данные временной таблицы «Оборудование» и скопируем фрагмент текста запроса к временной таблице (Рисунок 398-399).

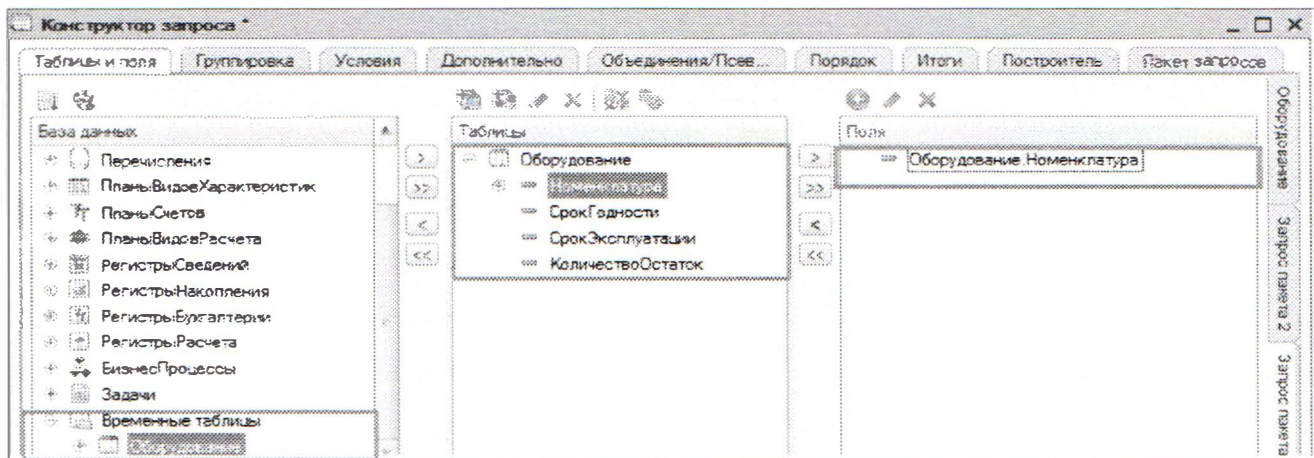


Рисунок 398

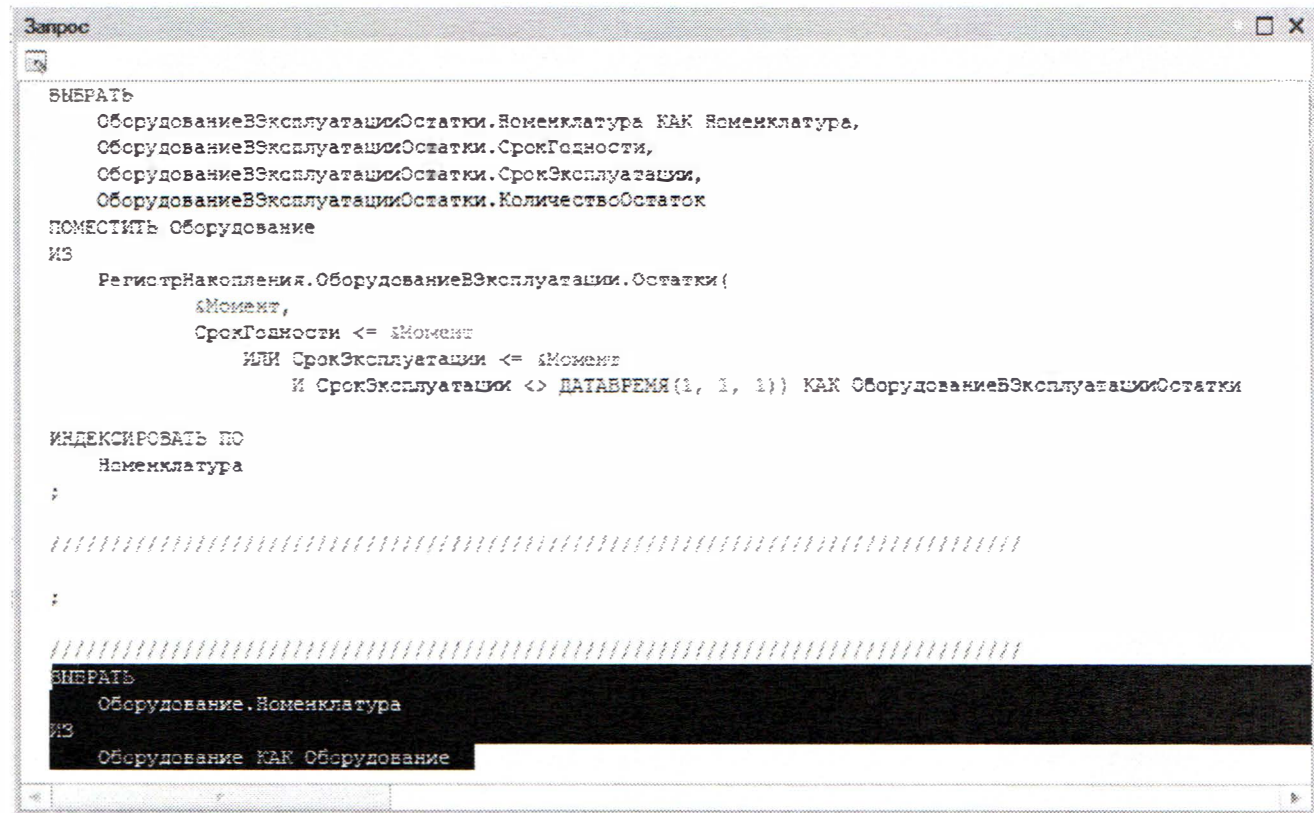


Рисунок 399

После этого удалим Запрос пакета3 и перейдем к параметрам условия Запроса пакета2 (Рисунок 400).

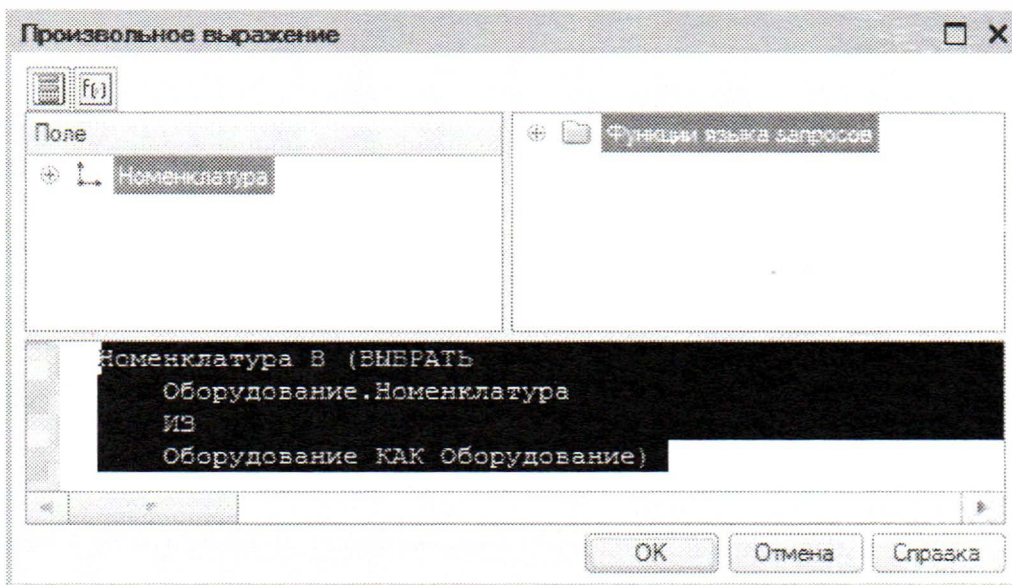


Рисунок 400

Подобные действия нужно выполнять, чтобы не рассчитывать все записи таблицы по остаткам, которых может быть тысячи, а несколько сузить область данных.

Однако, по условиям задачи, списывать необходимо оборудование, пока же мы работали с остатками. В связи с этим обратимся к временной таблице «Оборудование» и свяжем ее левым соединением с «ОстаткиНоменклатуры» по ранее проиндексированному полю «Номенклатура» (Рисунок 401-402).

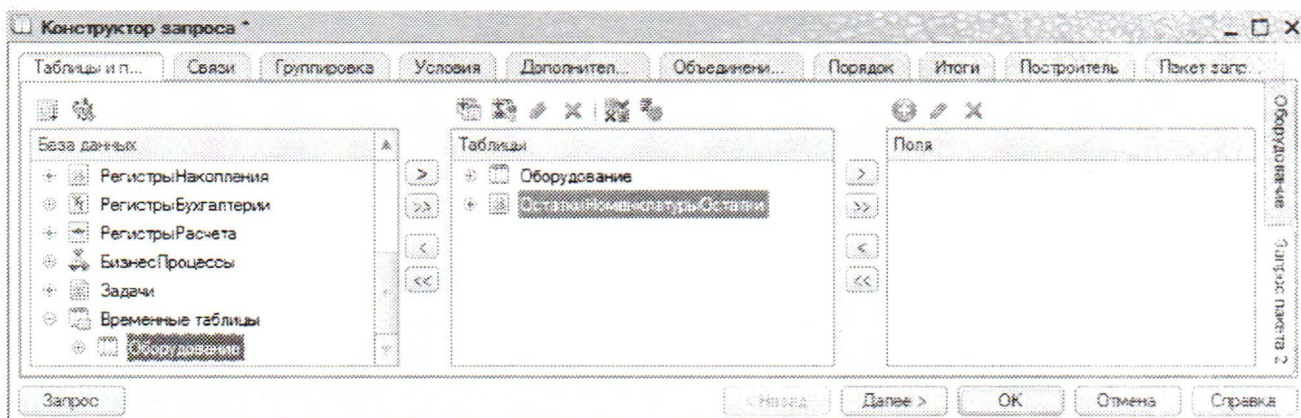


Рисунок 401

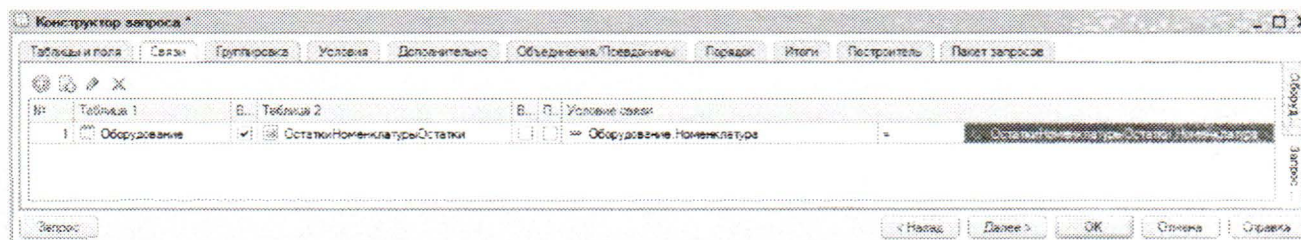


Рисунок 402

Определимся с составом полей: из таблицы «Оборудование» нас интересуют все поля; из таблицы «ОстаткиНоменклатуры» – «Количество» и «Сумма» (Рисунок 403).

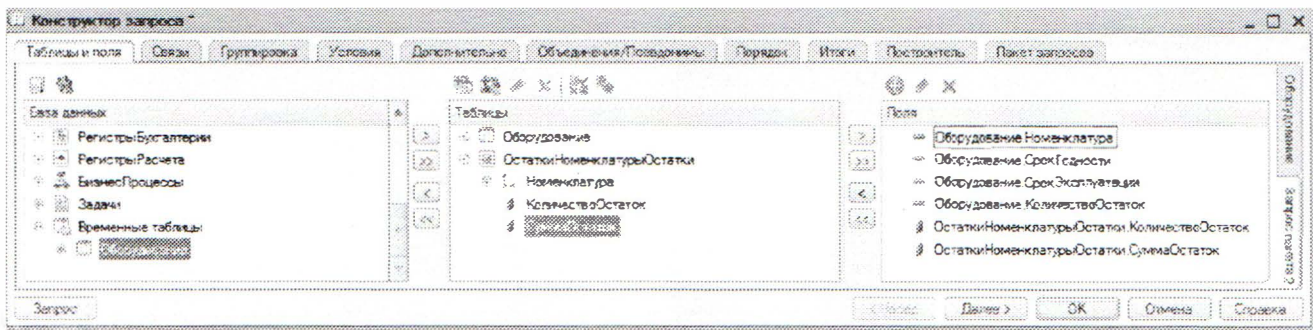


Рисунок 403

Количество на остатках всегда должно быть (поле «КоличествоОстаток» таблицы «ОстаткиНоменклатуры»), в противном случае – это говорит о том, что что-то идет неправильно. Тем не менее, проверка на «NULL» все равно должна быть. Поле «СуммаОстаток» таблицы «ОстаткиНоменклатуры» нам нужна для того, чтобы все корректно списать.

Так как мы имеем два поля с похожим названием в двух таблицах, на закладке «Объединения/псевдонимы» отредактируем наименования полей (Рисунок 404).

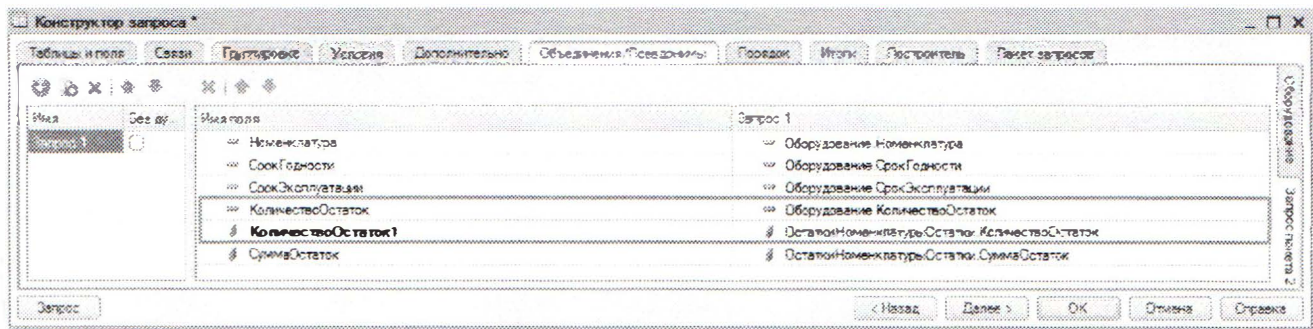


Рисунок 404

К наименованию «СуммаОстаток» добавим и удалим единицы для того чтобы его выделить жирным шрифтом – это требуется для последующего редактирования полей (Рисунок 405).

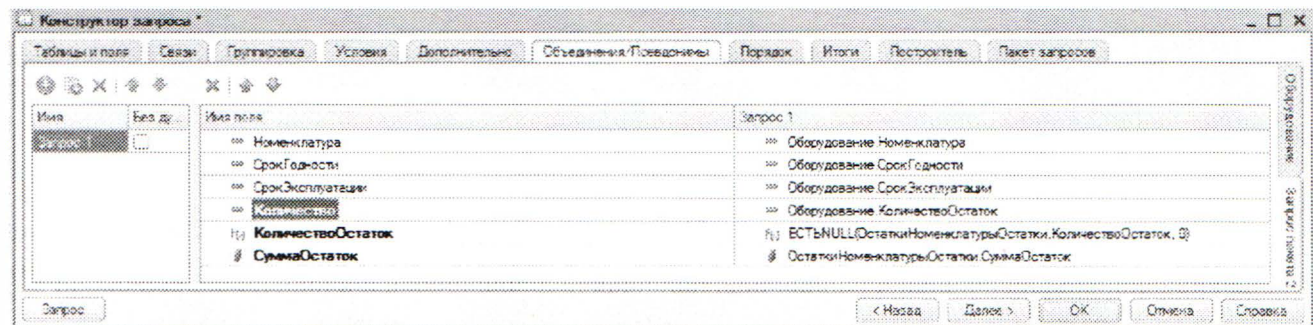


Рисунок 405

Далее требуется выполнить проверку на «NULL» в полях «Количество» и «Сумма» таблицы остатков номенклатуры (чтобы не сравнивать в дальнейшем значение данных полей с «NULL», воспользуемся функцией «ЕСТЬNULL» языка запросов) (Рисунок 406).

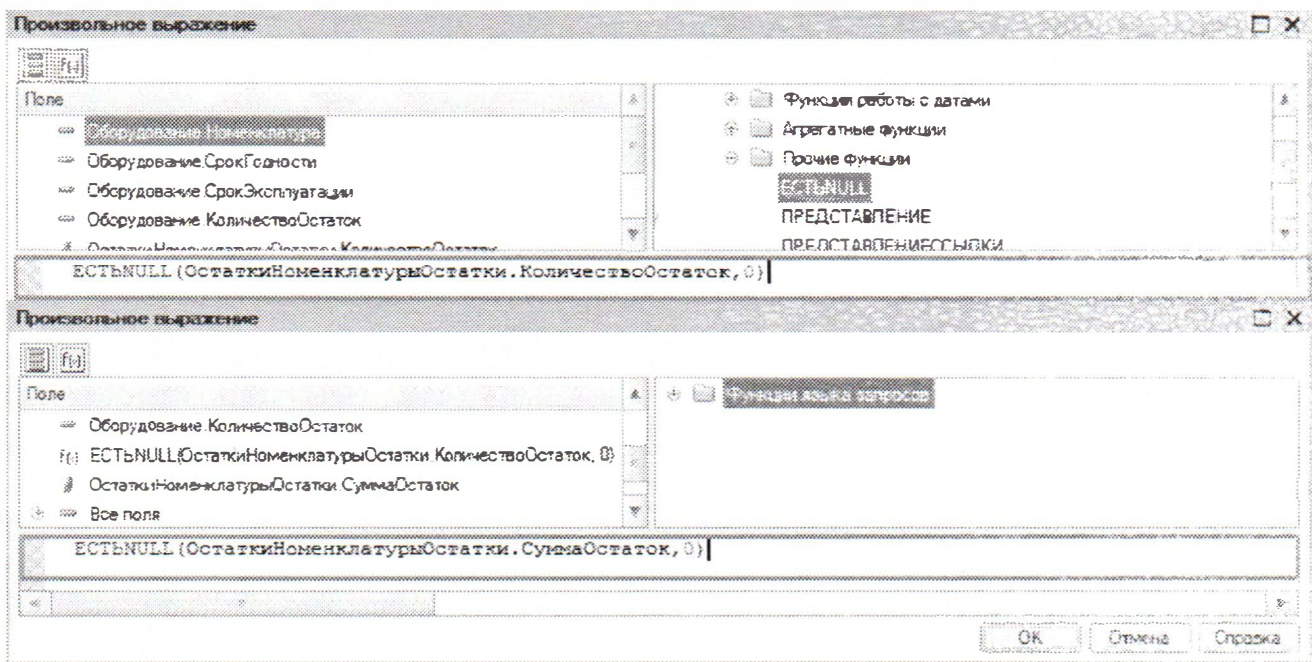


Рисунок 406

Посмотрим, что произойдет, когда мы свяжем эти таблицы: в таблице «Оборудование» строк «СрокГодности», «СрокЭксплуатации» будет множество для номенклатурной позиции, а в таблице остатков номенклатуры – каждой номенклатурной позиции будет соответствовать только одна строка с полями «КоличествоОстаток» и «СуммаОстаток». Следовательно, когда мы свяжем эти таблицы по номенклатуре, получим итоговую таблицу, где много раз будет встречаться одна и та же номенклатура, и в каждой из этих строк значение полей «КоличествоОстаток» и «СуммаОстаток» будет одинаковым.

Для того чтобы корректно вывести данные по номенклатуре, воспользуемся механизмом итогов языка запросов. Для этого на закладке «Итоги» в качестве группировочного поля выберем номенклатуру, а в строку итогов перенесем поля (Рисунок 407):

- *Количество* – значение из детальных записей таблицы «Оборудование», достаточно просуммировать;
- *КоличествоОстаток* – повторяющиеся значения для каждой номенклатуры таблицы «ОстаткиНоменклатуры», следовательно, суммировать не можем, используем функцию МАКСИМУМ;
- *СуммаОстаток* – повторяющиеся значения для каждой номенклатуры таблицы «ОстаткиНоменклатуры», следовательно, суммировать не можем, используем функцию МАКСИМУМ (с тем же успехом можно использовать функции МИНИМУМ, СРЕДНЕЕ, тогда, значения полей количества и суммы остатков номенклатуры будут повторяющимися).

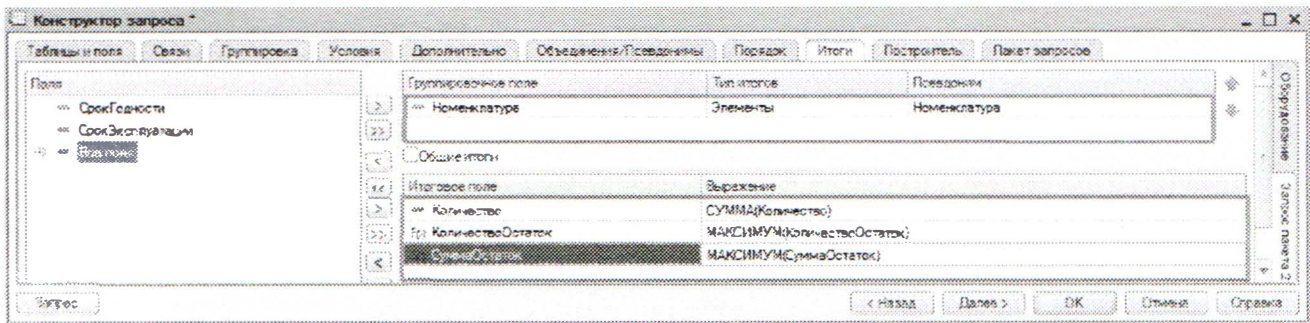


Рисунок 407

Теперь мы имеем все необходимые данные для списания оборудования и, что не менее важно, данные по сумме номенклатуры, которую необходимо списать уже с регистра остатков номенклатуры.

Итоговый текст запроса выглядит следующим образом:

"ВЫБРАТЬ

*ОборудованиеВЭксплуатацииОстатки.Номенклатура КАК Номенклатура,
ОборудованиеВЭксплуатации(Остатки.СрокГодности,
ОборудованиеВЭксплуатации(Остатки.СрокЭксплуатации,
ОборудованиеВЭксплуатации(Остатки.КоличествоОстаток*

ПОМЕСТИТЬ Оборудование

ИЗ

*РегистрНакопления.ОборудованиеВЭксплуатации(Остатки(
&Момент,
СрокГодности <= &Момент
ИЛИ СрокЭксплуатации <= &Момент*

И СрокЭксплуатации <> ДАТАВРЕМЯ(1, 1, 1)) КАК

ОборудованиеВЭксплуатации(Остатки

ИНДЕКСИРОВАТЬ ПО

Номенклатура

;

ВЫБРАТЬ

*Оборудование.Номенклатура КАК Номенклатура,
Оборудование.СрокГодности,
Оборудование.СрокЭксплуатации,
Оборудование.КоличествоОстаток КАК Количество,
«ЕСТЬNULL»(ОстаткиНоменклатуры(Остатки.КоличествоОстаток, 0) КАК*

КоличествоОстаток,

«ЕСТЬNULL»(ОстаткиНоменклатуры(Остатки.СуммаОстаток, 0) КАК

СуммаОстаток

ИЗ

Оборудование КАК Оборудование

ЛЕВОЕ СОЕДИНЕНИЕ

РегистрНакопления.ОстаткиНоменклатуры.Остатки(

&Момент,

Номенклатура В

(ВЫБРАТЬ

Оборудование.Номенклатура

ИЗ

Оборудование КАК Оборудование)) КАК

ОстаткиНоменклатурыОстатки

ПО Оборудование.Номенклатура =

ОстаткиНоменклатурыОстатки.Номенклатура

ИТОГИ

СУММА(Количество),

МАКСИМУМ(КоличествоОстаток),

МАКСИМУМ(СуммаОстаток)

ПО

Номенклатура"

Акцентируем внимание, что в качестве периода параметра запроса в данном случае передана «Дата», а не «МоментВремени()», так как мы будем работать не со значением «Дата+СсылкаНаДокумент», а просто с датой документа (Рисунок 408).

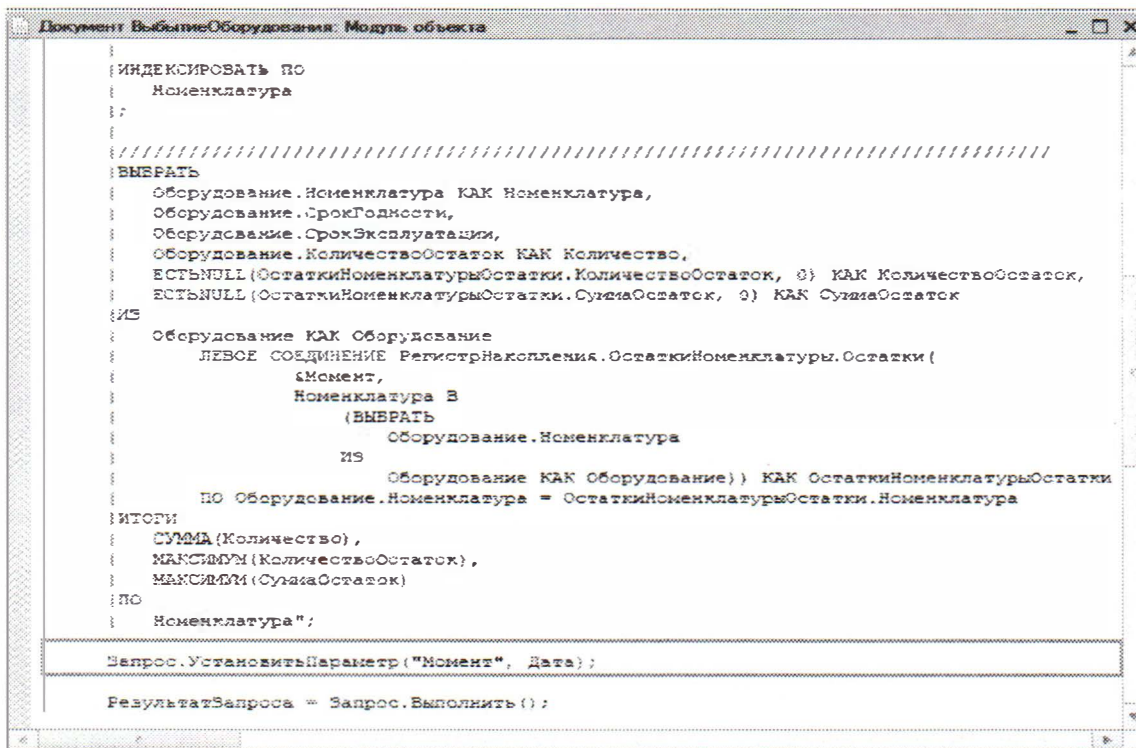


Рисунок 408

5.1.4 Метод дополнения.

Рассмотрим одну из особенностей таблицы регистра накопления.

Для этого запустим приложение в режиме «1С:Предприятие» и запустим консоль запросов (Рисунок 409).

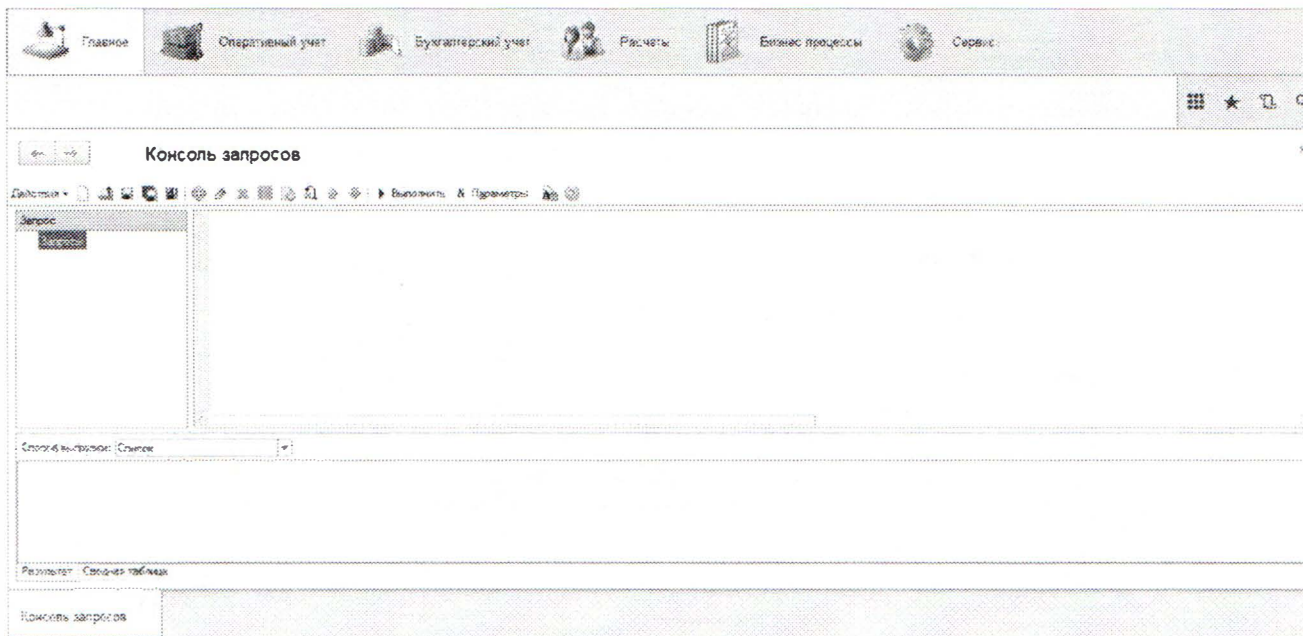


Рисунок 409

В данных информационной базы есть документ «Ввод в эксплуатацию 000000001 от 20.01.2014 10:46:24», записи по которой и получим в консоли запроса (Рисунок 410).

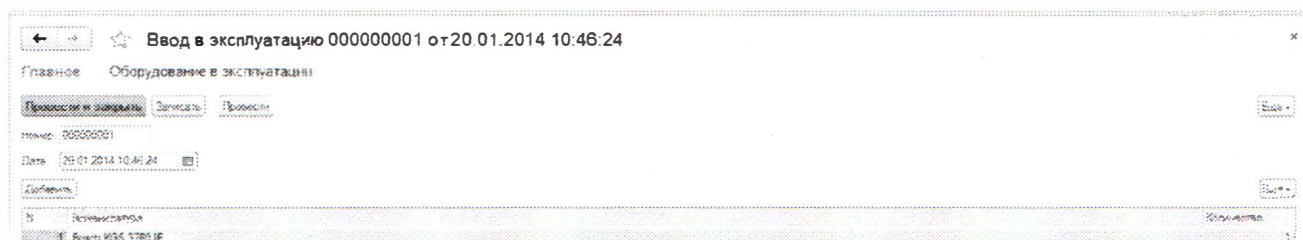


Рисунок 410

Обратимся из консоли запроса к виртуальной таблице остатков регистра накопления «ОборудованиеВЭксплуатации» и посмотрим, какие параметры она имеет (Рисунок 411).

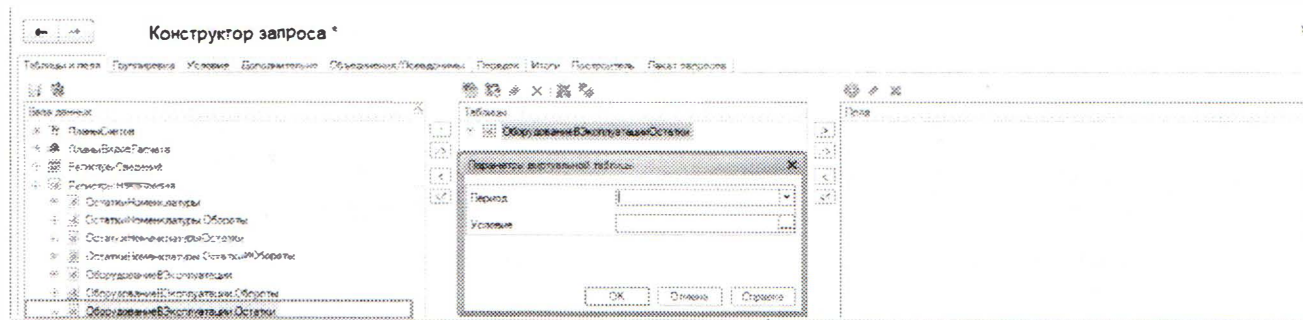


Рисунок 411

Это стандартные параметры виртуальной таблицы, с которыми мы имели дело ранее. Иначе обстоит дело с виртуальной таблицей остатков и оборотов (Рисунок 412).

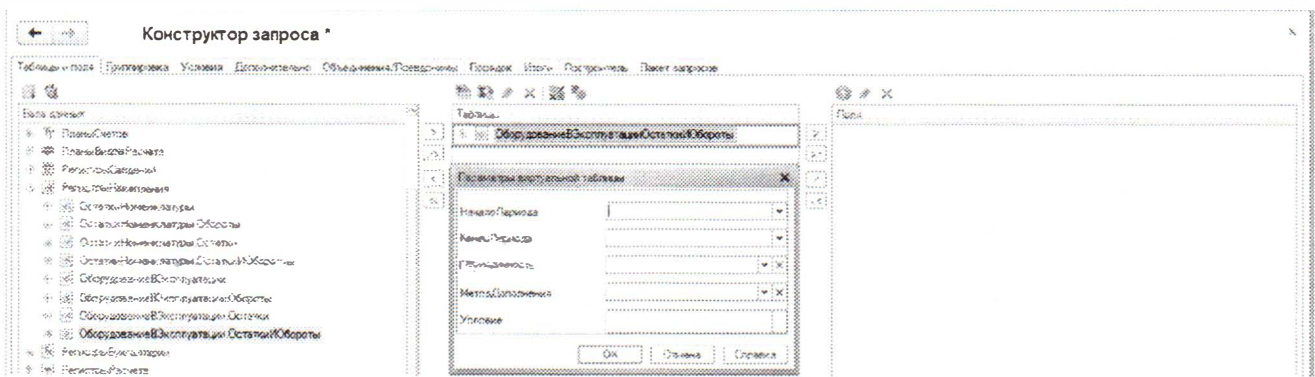


Рисунок 412

Следует разобраться с параметром «МетодДополнения» и его типами: (Рисунок 413):

1. Движения;
2. ДвиженияИГраницыПериода.

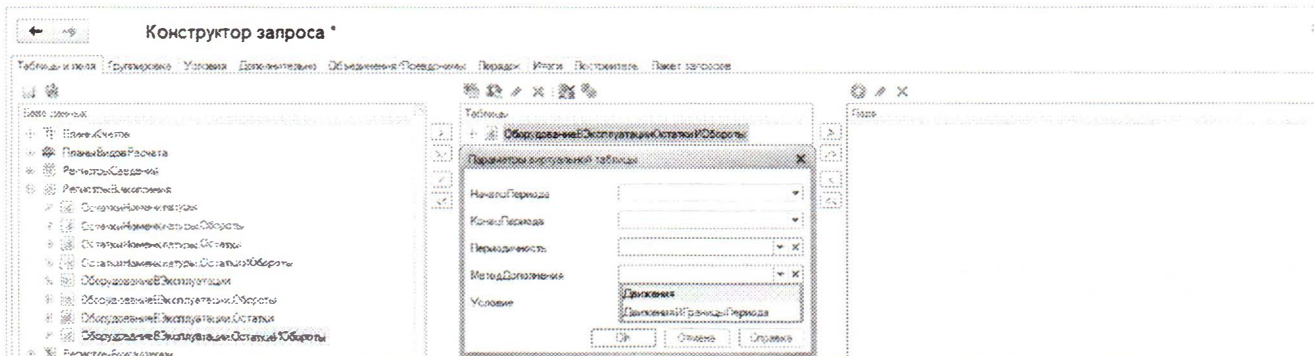


Рисунок 413

Для просмотра механизма «МетодаДополнения» следует поставить параметр «Периодичность» в значение «Регистратор» и выбрать все поля таблицы (Рисунок 414).

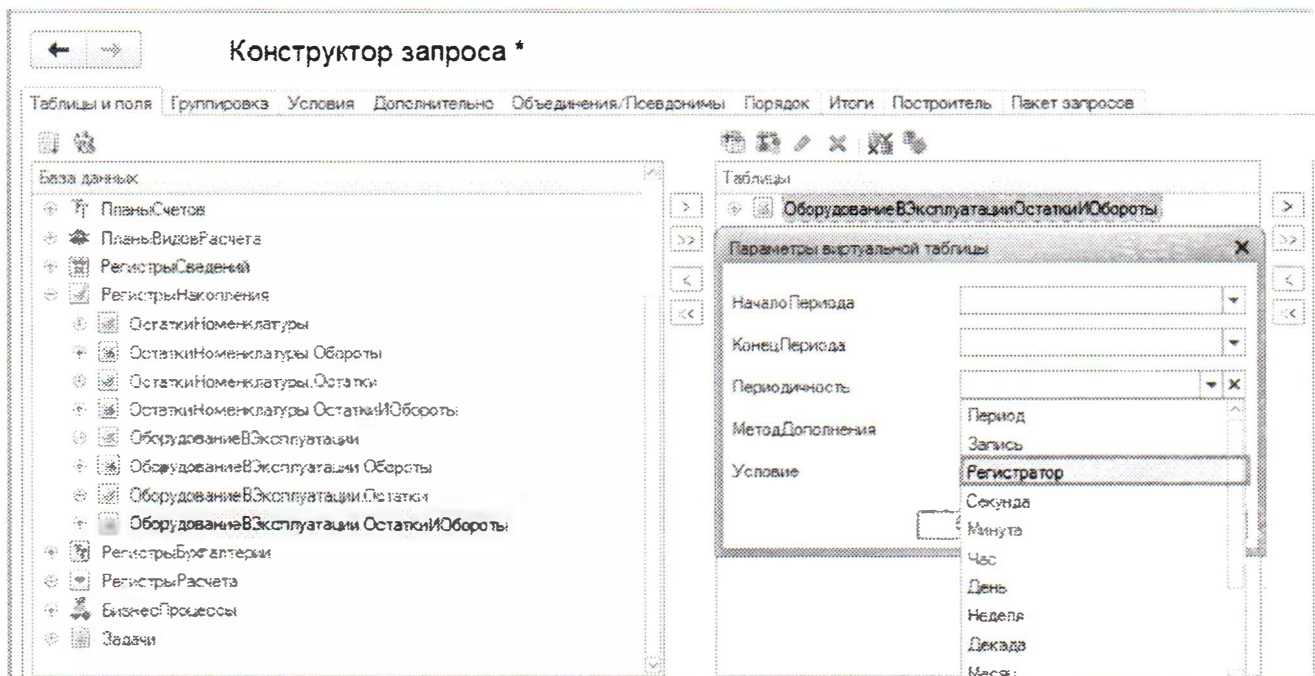


Рисунок 414

В качестве НачалаПериода укажем ДатуНачала; в качестве КонцаПериода – ДатуОкончания (Рисунок 415).

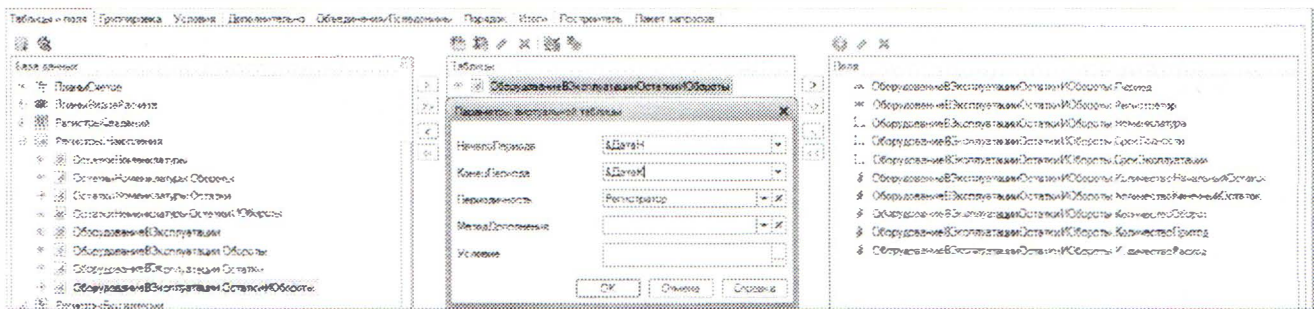


Рисунок 415

Сохраним запрос и перейдем в консоль запроса, в которой нажмем кнопку «Параметры» (Рисунок 416).

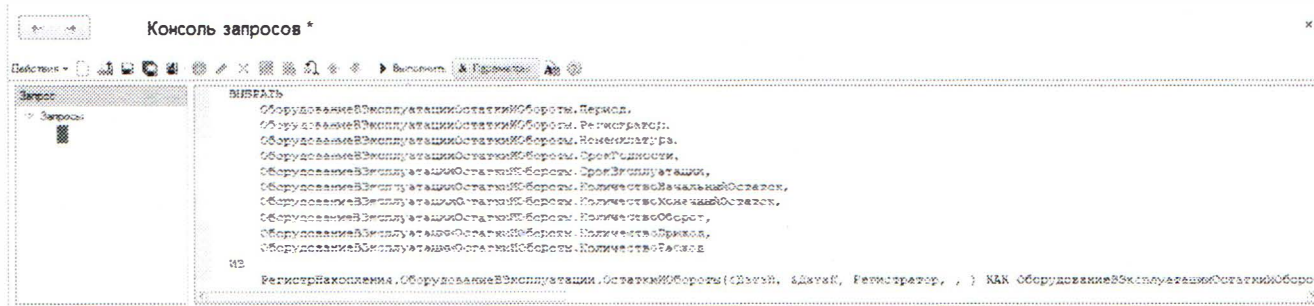


Рисунок 416

В форме параметров выполним команду «Получить из запроса» и в качестве параметров зададим даты (Рисунок 417):

- 19.01.2014 00:00 – ДатаН;
- 21.01.2014 00:00 – ДатаК.

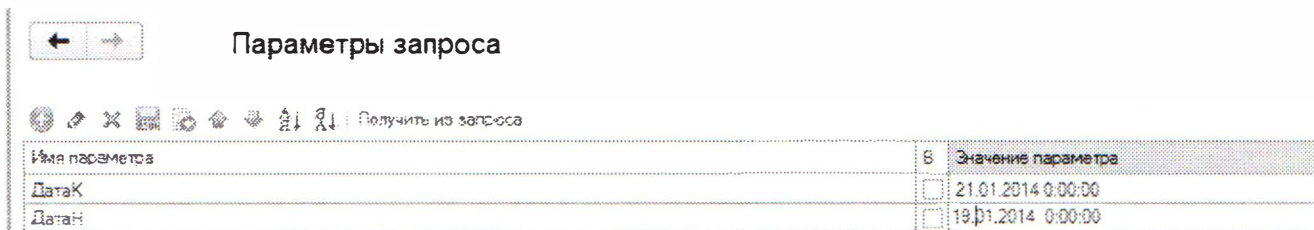


Рисунок 417

Для проверки результата выполним запрос (Рисунок 418):

ВЫБРАТЬ

*ОборудованиеВЭксплуатацииОстаткиИОбороты.Период,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.Регистратор,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.Номенклатура,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.СрокГодности,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.СрокЭксплуатации,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.КоличествоНачальныйОстаток,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.КоличествоКонечныйОстаток,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.КоличествоОборот,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.КоличествоПриход,
 ОборудованиеВЭксплуатацииОстаткиИОбороты.КоличествоРасход*

ИЗ

РегистрНакопления.ОборудованиеВЭксплуатации.ОстаткиИОбороты(&ДатаН, &ДатаК,
Регистратор,) КАК ОборудованиеВЭксплуатацииОстаткиИОбороты

Период	Регистратор	Номернакопления	Остаток	Сторг	КоличествоналичныйЭкспорт	КоличествооплатывЭкспорт	Количество	КоличествоПродаж	КоличествоПлатеж
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00

Рисунок 418

В результате выполнения запроса мы получили набор строк.

Зададим параметр запроса МетодДополнения, тип – «Движения» и выполним запрос (Рисунок 419).

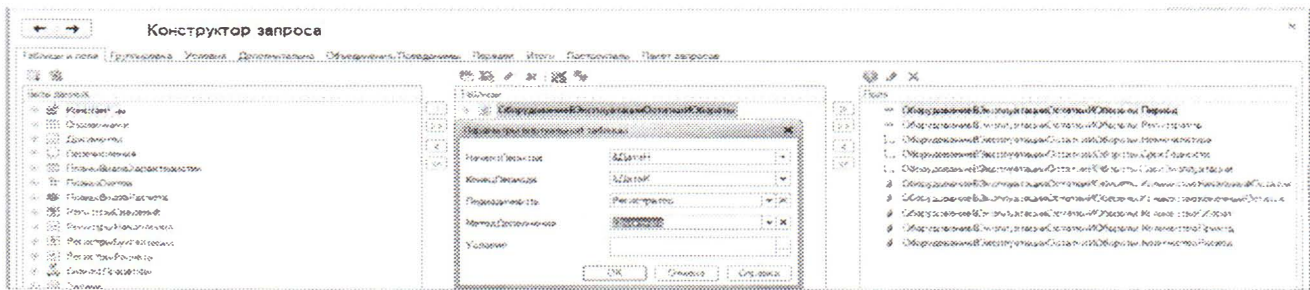


Рисунок 419

Проанализируем результат запроса (Рисунок 420).

Период	Регистратор	Номернакопления	Остаток	Сторг	КоличествоналичныйЭкспорт	КоличествооплатывЭкспорт	Количество	КоличествоПродаж	КоличествоПлатеж
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00
01.01.2014-01.01.2014	Транспортное средство	Валент. КС 3760 И	1,42	0,00	0,00	0,00	0,00	0,00	0,00

Рисунок 420

После выбора параметра «МетодДополнения» с типом «Движения» в результат запроса попали только те записи, у которых задан регистратор, состав колонок не изменился.

Зададим в качестве параметра «МетодаДополнения» значение «ДвиженияИГраницыПериода» (Рисунок 421).

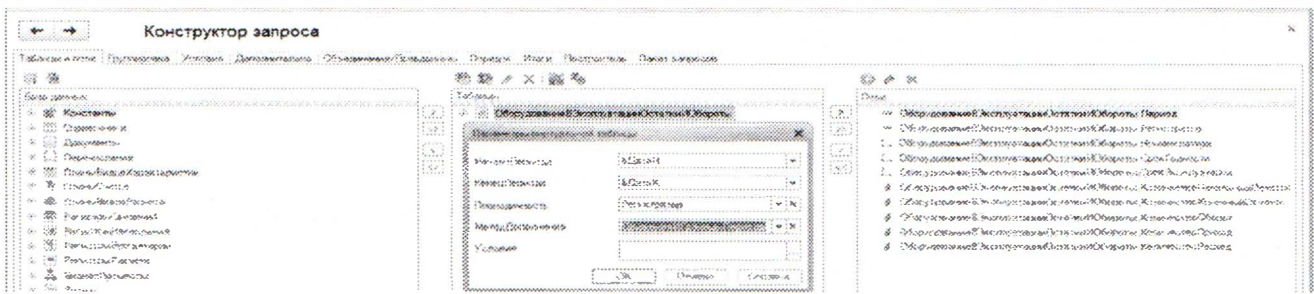


Рисунок 421

В результате выполнения запроса получим еще и записи с пустым регистратором, по начальным и конечным остаткам на начало периода и конец периода, в данном случае – по каждой номенклатурной позиции (Рисунок 422).

Период	Наименование	Склад	Склад 2	Количество	Количество	Количество	Количество	Количество
19.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02	19.02				
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02	15.02				
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					
31.01.2014 00:00	Продукция извлечения	Склад №3 3790 R	12.02					

Рисунок 422

Однако записи по началу периода в таблице результатов мы не видим. Это связано с тем, что на начало периода (19.01.2014 00:00) документов в информационной базе еще не было введено, таким образом, в результат попали данные по конечным остаткам. Если бы мы задали другие границы периода, то получили бы и начальные остатки.

5.2 Бухгалтерский учет.

5.2.1 Бухгалтерский учет. Списание комплектов и комплектующих.

Рассмотрим следующую задачу, ее формулировка представлена ниже (Рисунок 423).

Необходимо создать документ «Операция», с помощью которого пользователь должен иметь возможность ввести проводки с произвольной корреспонденцией счетов. При решении задач следует учитывать возможность наличия проводок, сформированных с помощью данного документа.

Требуется организовать возможность учета продажи комплектов. В системе должна храниться информация о составе комплекта (какие номенклатурные позиции и в каком количестве входят в данный комплект). В состав комплекта не могут входить другие комплекты. Следует обеспечить уникальность деталей в рамках комплектов, т.е. одна и та же деталь не должна входить в состав разных комплектов.

Операция сборки как таковая в компании не осуществляется. Когда покупатель приходит получать комплект ответственный сотрудник компании в соответствии со спецификацией отгружает необходимое количество комплектующих.

Продажи осуществляются документом «Расходная накладная». В табличной части документа указываются комплекты и их количество.

Документ реализует следующие проводки:

Дт «Прибыли и убытки» - Кт «Товары»

на количество и себестоимость списываемых комплектующих. В случае продажи комплекта необходимо списать его комплектующие согласно составу комплекта

Дт «Покупатели» - Кт «Прибыли и убытки» на сумму в продажных ценах.

Себестоимость рассчитывается как средняя по номенклатурной позиции на складе.

Рисунок 423

Рассмотрим структуру метаданных конфигурации, созданную для решения этой задачи. В подсистеме бухгалтерского учета нас, прежде всего, интересуют объекты, которые реализуют возможность продажи комплекта:

– Справочник «Номенклатура» имеет признак «Комплект» со значением типа булево, который определяет принадлежность номенклатуры составу комплекта (общий состав реквизитов справочника был рассмотрен в контексте решения задачи оперативного учета) (Рисунок 424).

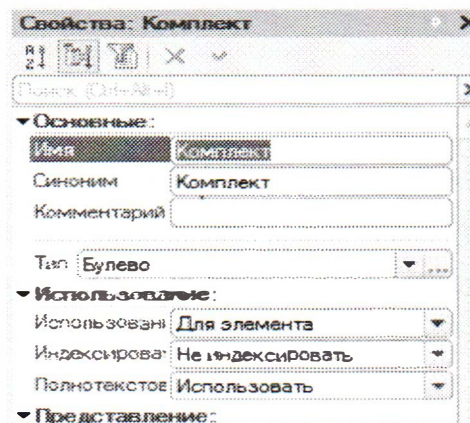


Рисунок 424

– Регистр сведений «Состав комплектов» хранит информацию по комплектам (номенклатура в составе комплекта, количество) (Рисунок 425).

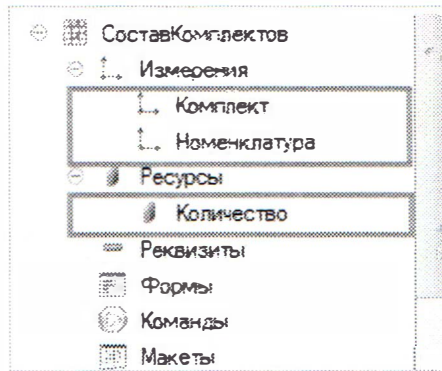


Рисунок 425

«Состав комплектов» является периодическим регистром, следовательно, состав комплектов может изменяться (Рисунок 426).

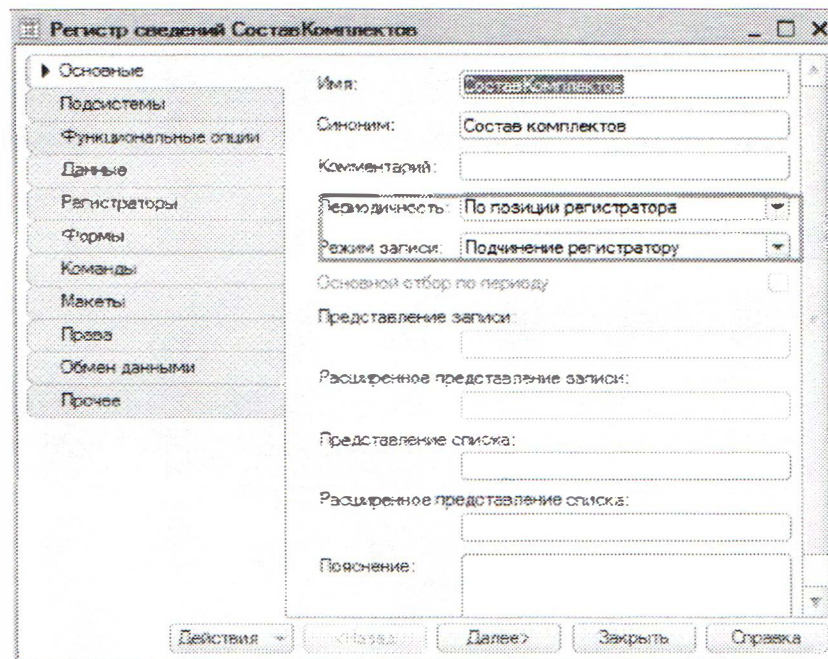


Рисунок 426

– Документ «Изменение состава комплектов» регистрирует номенклатуру в составе комплекта, соответственно, содержит табличную часть «Состав» с колонками: «Номенклатура», «Количество». При проведении документ делает движения по регистру «Состав комплектов» (Рисунок 427).

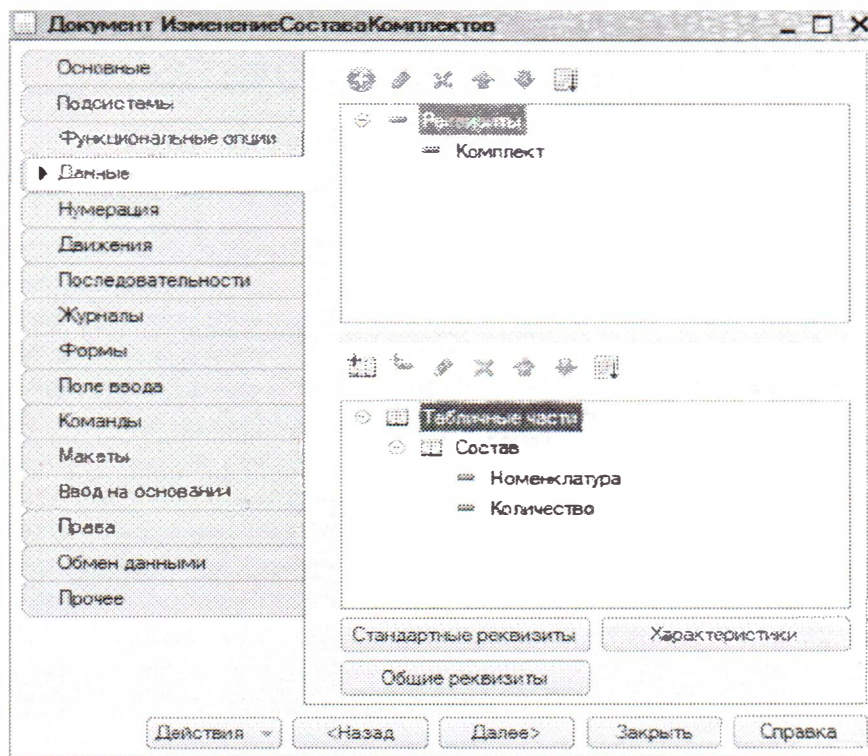


Рисунок 427

– План счетов «Управленческий» содержит счета учета, которые используются в момент приходования номенклатуры (Рисунок 428).

Имя	Код	Наименование	Вид	З...	Порядок	К.	Се...	Субконто 1	Субконто 2
Активы	01	Активы	Активный		01				
Касса	01.01	Касса	Активный		01.01				
Товары	01.02	Товары	Активный		01.02	✓		Номенкла...	Склад
Товары в Пути	01.03	Товары в пути	Активный		01.03				
Покупатели	01.04	Покупатели	Активн...		01.04		✓	Номенкла...	
Основное Производство	01.05	Основное производство	Активный		01.05				
Общехозяйственные Затраты	01.06	Общехозяйственные за...	Активный		01.06				
Кредиты и Займы Выданные	01.07	Кредиты и займы выда...	Активный		01.07				
Материалы	01.08	Материалы	Активный		01.08				
Обязательства	02	Обязательства	Пассив...		02				
Сотрудники	02.01	Сотрудники	Активн...		02.01				
Поставщики	02.02	Поставщики	Активн...		02.02				
Кредиты и Займы Полученные	02.03	Кредиты и займы получ...	Пассив...		02.03				
Налоги	02.04	Налоги	Пассив...		02.04				
Прибыли и Убытки	03	Прибыли и убытки	Активн...		03				

Рисунок 428

Из обработки проведения Приходной накладной можно увидеть, что приход происходит на счет «Товары» и списание со счета «Поставщики» с аналитикой по складам (Рисунок 429).

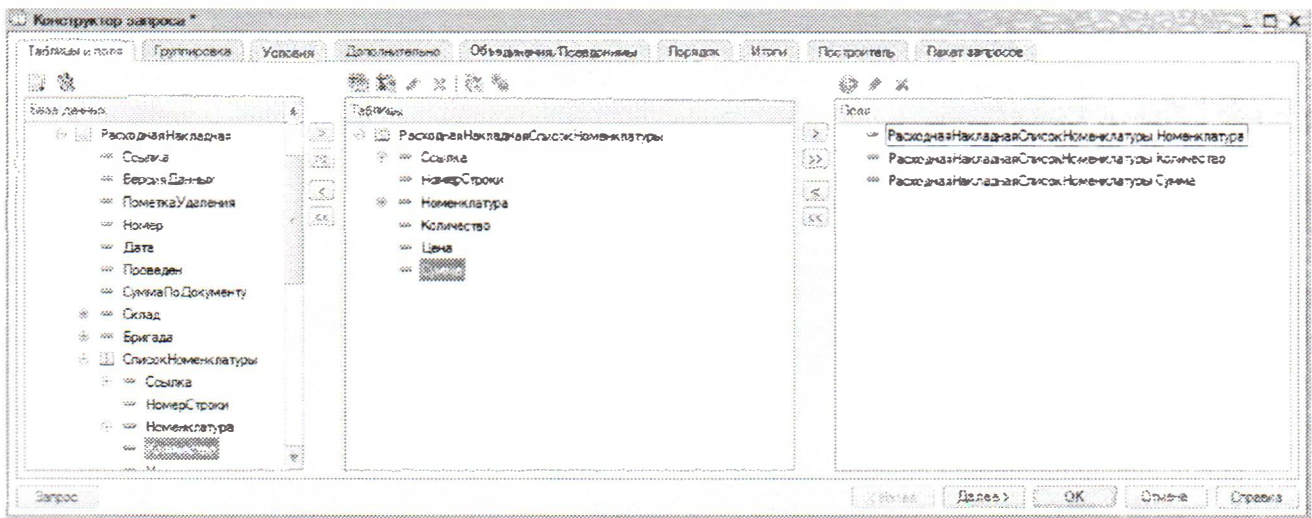


Рисунок 431

Сгруппируем записи по номенклатуре, просуммировав значения количества и суммы (Рисунок 432).

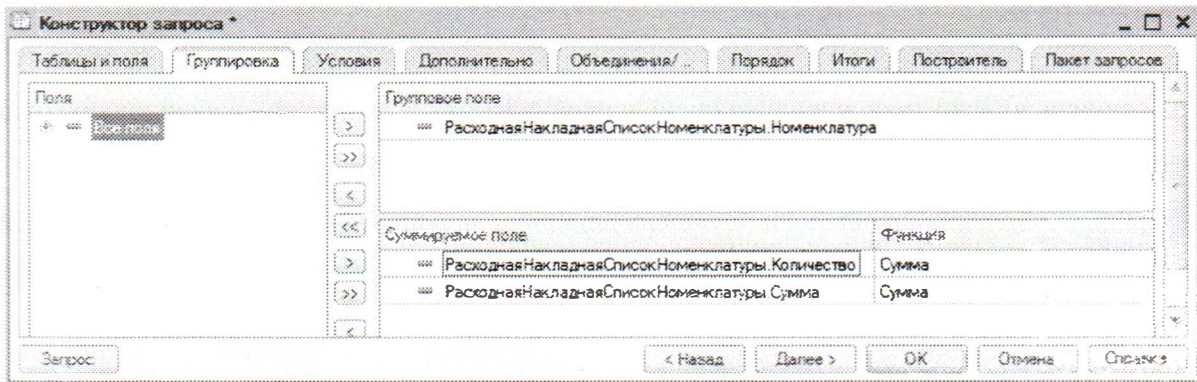


Рисунок 432

Добавим условие по ссылке на документ, чтобы данные выводились не по всем документам в информационной базе, а только по интересующим нас (Рисунок 433).

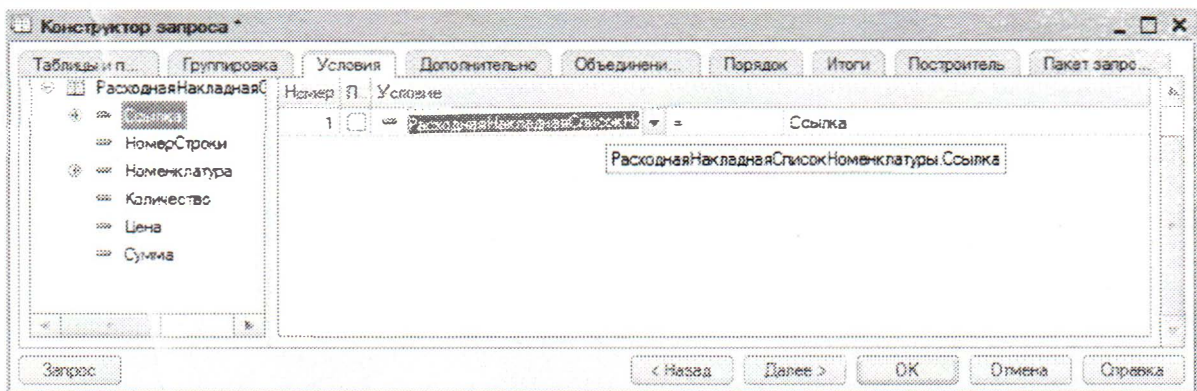


Рисунок 433

Далее поместим все данные во временную таблицу «Данные ТЧ» (Рисунок 434).

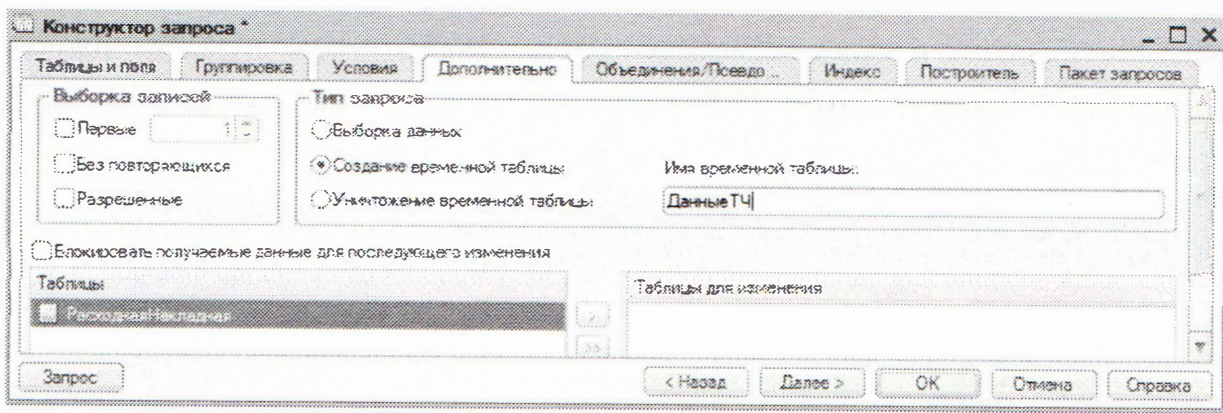


Рисунок 434

Проиндексируем записи по номенклатуре, так как по этому реквизиту в дальнейшем мы будем связывать таблицы (Рисунок 435).

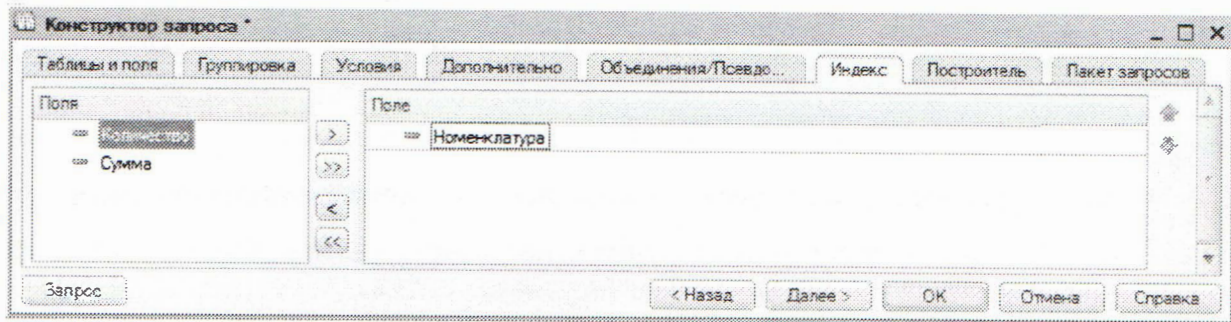


Рисунок 435

Таким образом, мы получили данные табличной части документа, которыми будем оперировать. На закладке «Пакет запросов» создадим запрос пакета 2, задача разработки которого – понять, какая часть из номенклатурных позиций табличной части является комплектом. Для этого необходимо, прежде всего, получить все комплектующие, которые находятся в регистре сведений «СоставКомплектовСрезПоследних» на момент проведения документа (Рисунок 436).

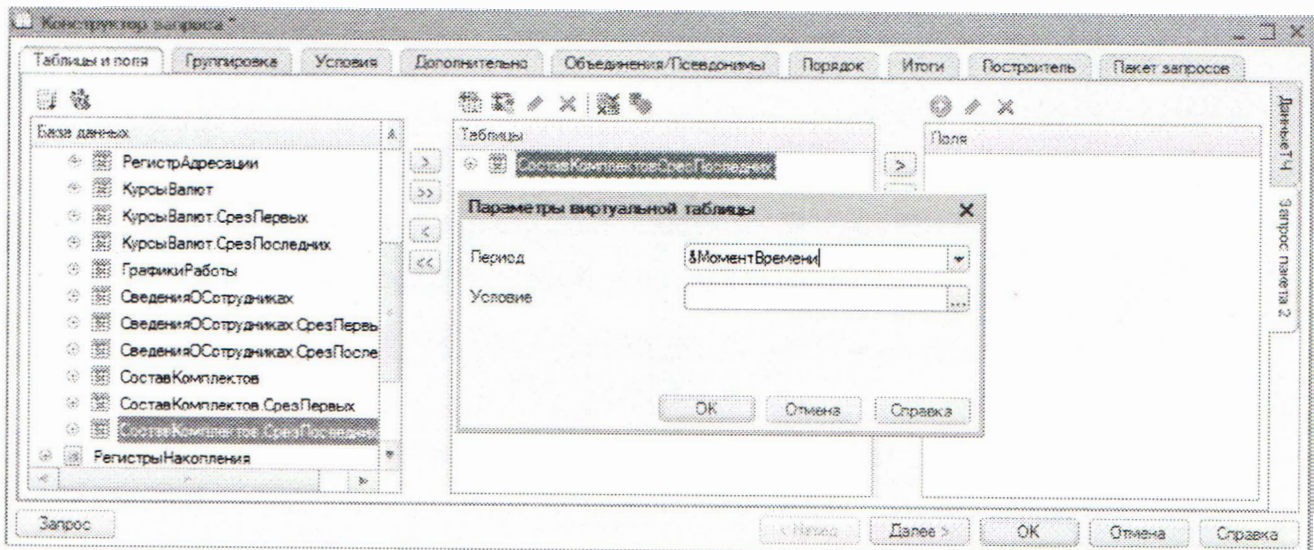


Рисунок 436

В параметре условия нам нужно отобрать только те значения, которые необходимо рассчитывать, для этого воспользуемся тем же методом, что и ранее – с использованием

пакетного запроса, который, после получения условия, будет удален. Для этого создадим еще один запрос пакета и получим в нем данные из временной таблицы «ДанныеТЧ» (Рисунок 437).

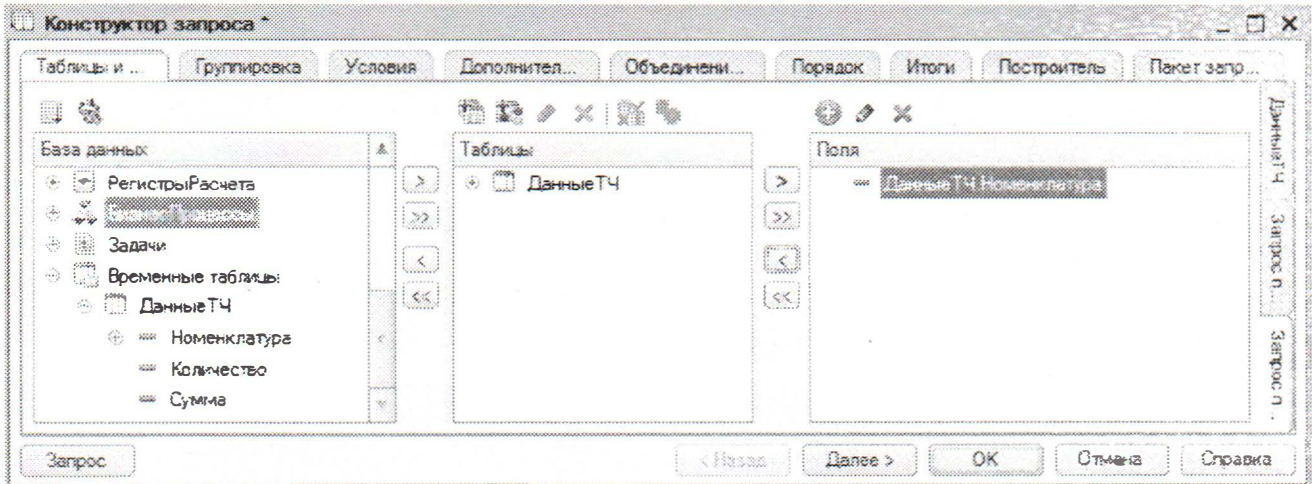
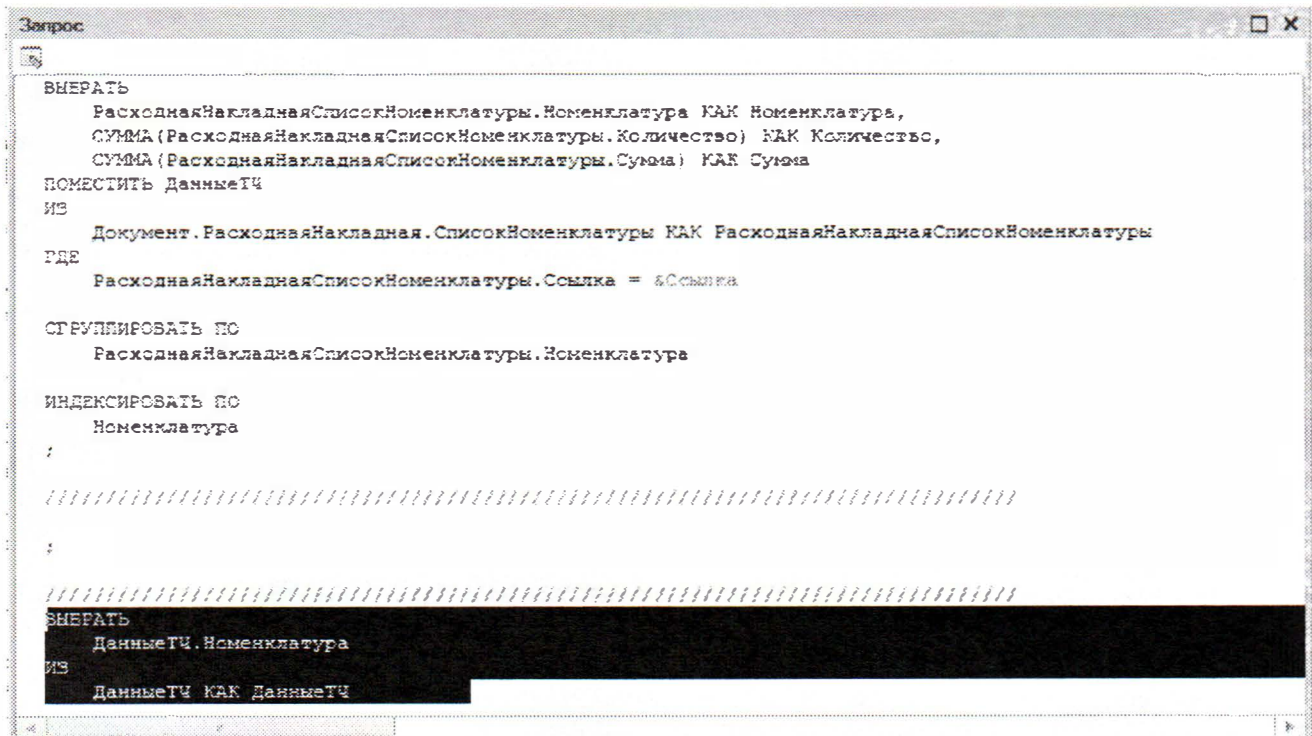


Рисунок 437

Скопируем текст последнего запроса пакета и вставим его в параметр «Условие» таблицы «СоставКомплектаСрезПоследних», после чего удалим запрос пакета 3 (Рисунок 438-439).



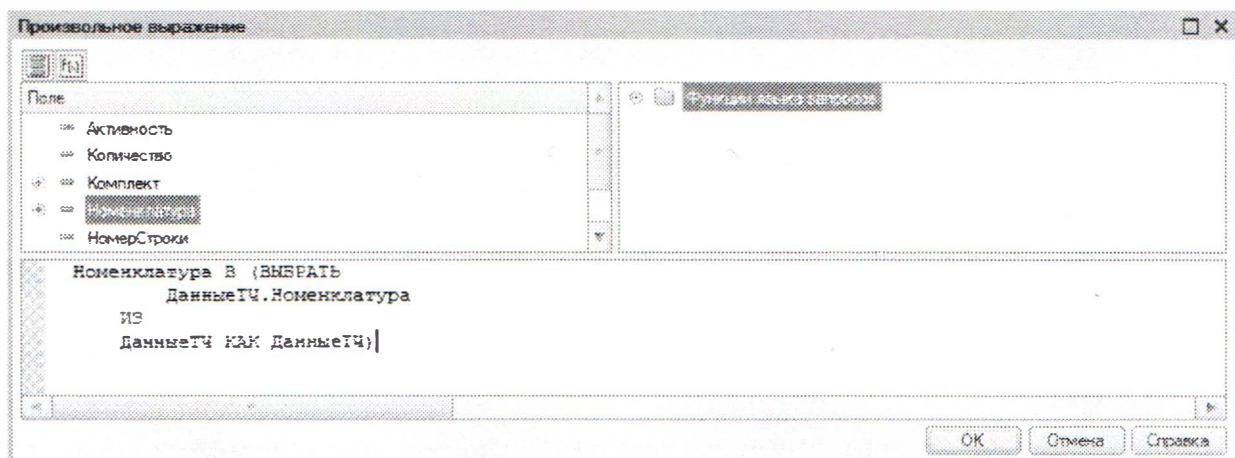


Рисунок 439

Следует отметить, что подобные действия ускоряют получение результата запроса при условии наличия в информационной базе большого количества записей.

Следующим шагом нужно связать таблицы данных табличной части с таблицей среза последних состава комплектов. Для этого на закладке конструктора запроса «Таблицы и поля» переместим из области «База данных» в область «Таблицы» временную таблицу «ДанныеТЧ» (Рисунок 440).

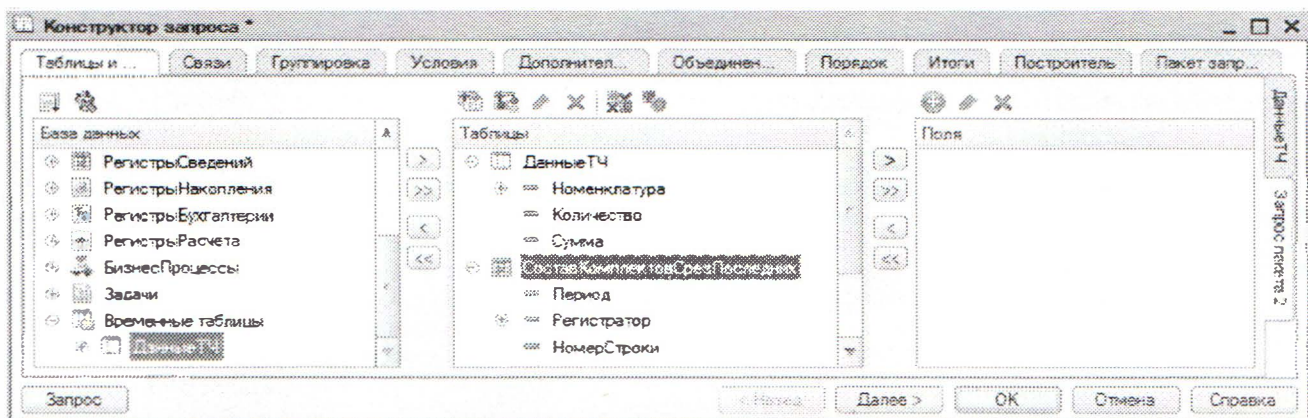


Рисунок 440

Далее следует проанализировать, каким способом данные таблицы связать. Так как мы берем данные из состава комплектов для того чтобы узнать, какие записи табличной части расходной являются комплектом, в качестве «Таблицы1» на закладке «Связи» следует выбрать таблицу «СоставКомплектовСрезПоследних»; «Таблицы2» – «ДанныеТЧ». В качестве условия связи в первом случае будет «Комплект», во втором – «Номенклатура» (таким образом, мы посмотрим в табличной части, какая номенклатура является комплектом).

Используем внутреннее соединение, так как здесь нам нужен только состав комплектующих (если номенклатурная позиция комплектующим не является, то эти данные в выборку не попадают, и обратная ситуация: если в составе комплектующих нет данных из табличной части, то эти данные нам так же не интересны). Таким образом, мы отобрали

только то, что есть в данных табличной части и является комплектом. Этому удовлетворяет условие соединения (Рисунок 441).

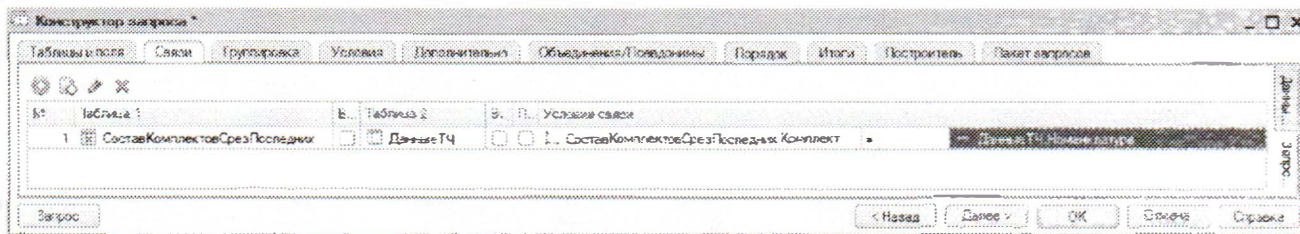


Рисунок 441

Далее следует определить состав полей. Из таблицы «СоставКомплектовСрезПоследних» нас интересуют прежде всего номенклатура и количество (Рисунок 442).

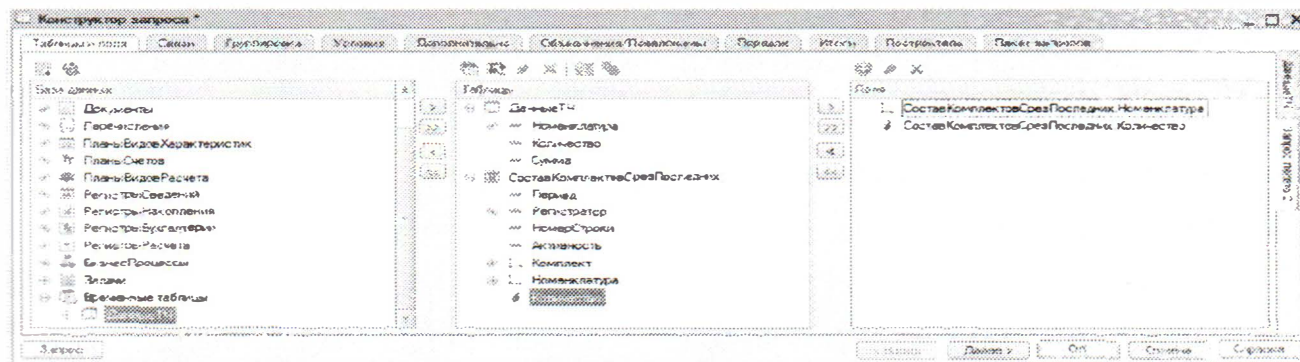


Рисунок 442

«ЕСТЬNULL» в данных значениях не используем, поскольку условие по типу связи внутреннего соединения отбирает только те данные, которые присутствуют в обеих таблицах. Что касается поля «СоставКомплектовСрезПоследних.Количество», то его значение показывает количество при расчете на одну единицу номенклатуры, но в таблице может содержаться любое положительное количество комплектов, следовательно, необходимо это количество умножить на количество из данных табличной части (Рисунок 443).

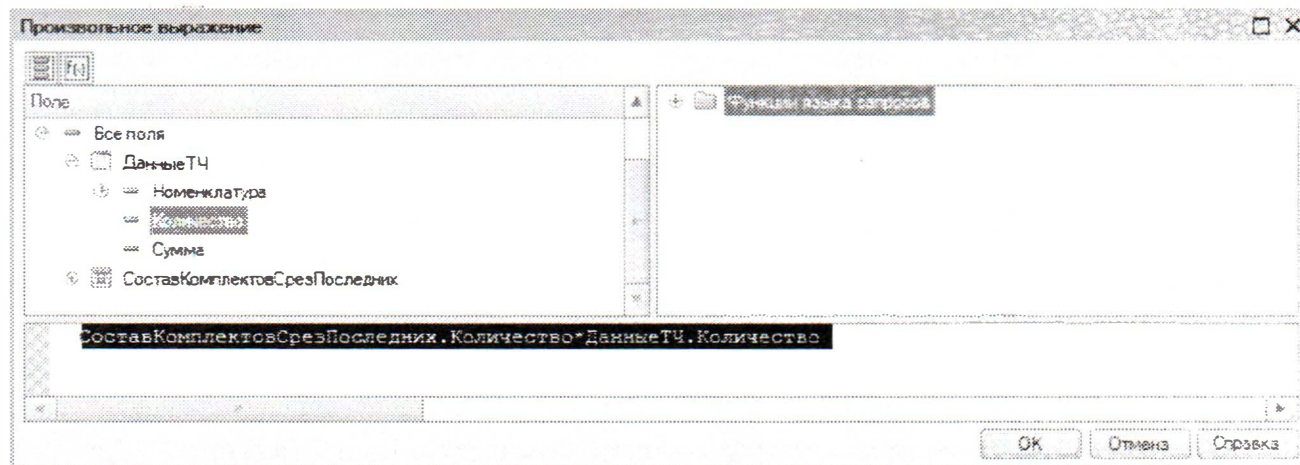


Рисунок 443

На закладке «Объединения/псевдонимы» конструктора запроса переименуем это поле в «Количество» (Рисунок 444).

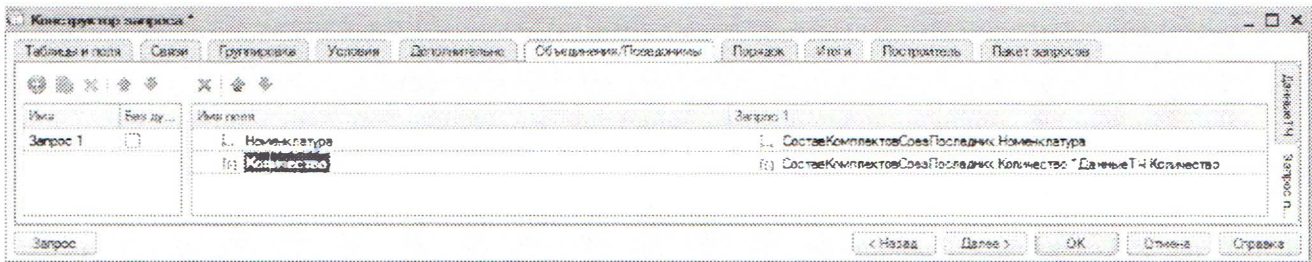


Рисунок 444

Далее поместим данные во временную таблицу «Комплектующие» (Рисунок 445).

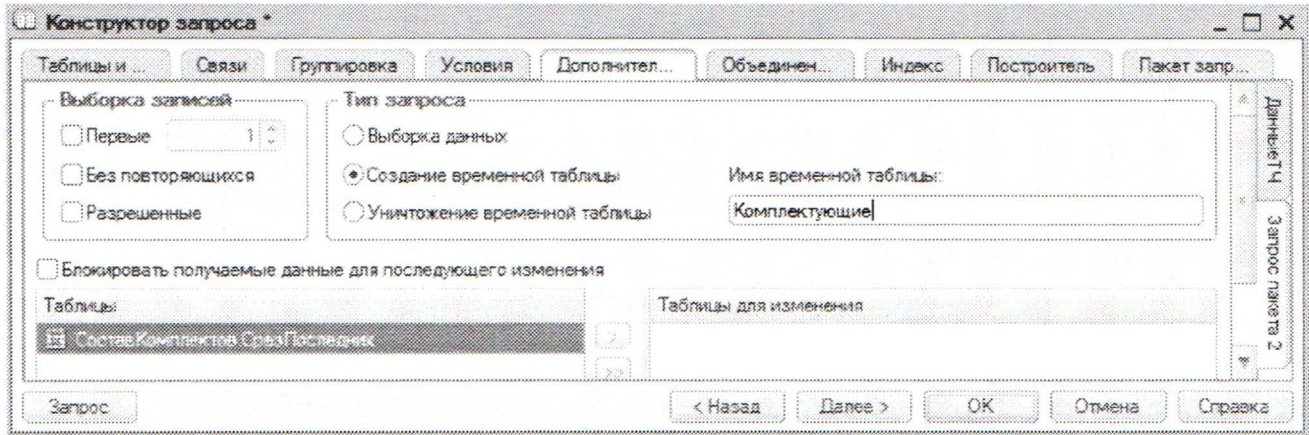


Рисунок 445

Таким образом, в составе таблицы «Комплектующие» получаем данные по количеству и номенклатуре, которая входит в состав комплекта. Но у нас могут быть и записи по номенклатуре, которая в состав комплекта не входит и продается напрямую. Их также следует добавить, чтобы потом корректно рассчитывать себестоимость. Для этого нам нужны данные по всей табличной части (для того, чтобы мы могли сделать проводку по счетам «Покупатели» и «Прибыль») и данные по всем комплектующим, которые не являются комплектами по самой номенклатуре из данных табличной части расходной (для корректного списания себестоимости). Для этого нам требуется в таблицу «Комплектующие» добавить поля по номенклатуре, которая не является комплектом.

На закладке «Объединения/Псевдонимы» добавим еще один запрос в составе временной таблицы «Комплектующие» (Рисунок 446).

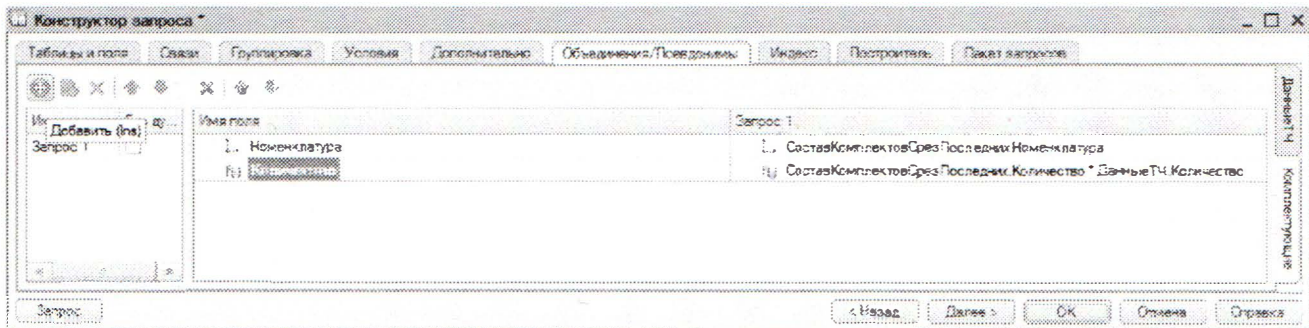


Рисунок 446

Из временной таблицы «ДанныеТЧ» нас интересуют номенклатура и количество (Рисунок 447).

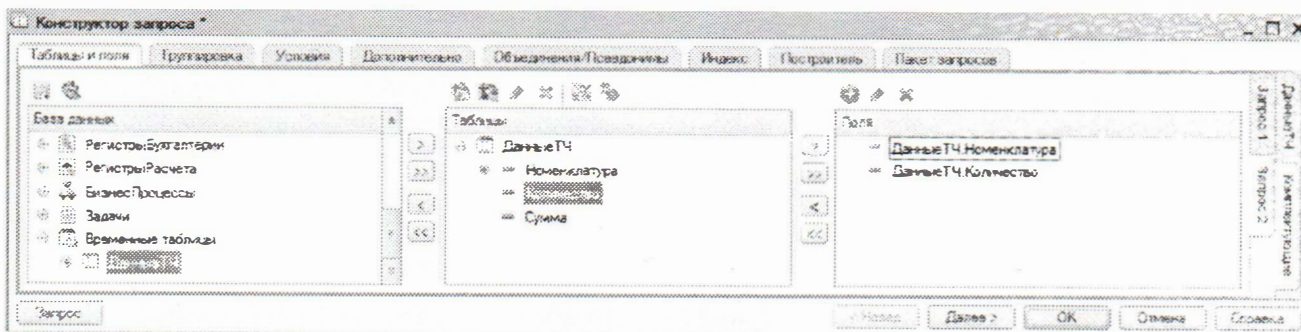


Рисунок 447

На закладке «Объединения/псевдонимы» назначим значения полям с отсутствующим значением (Рисунок 448).

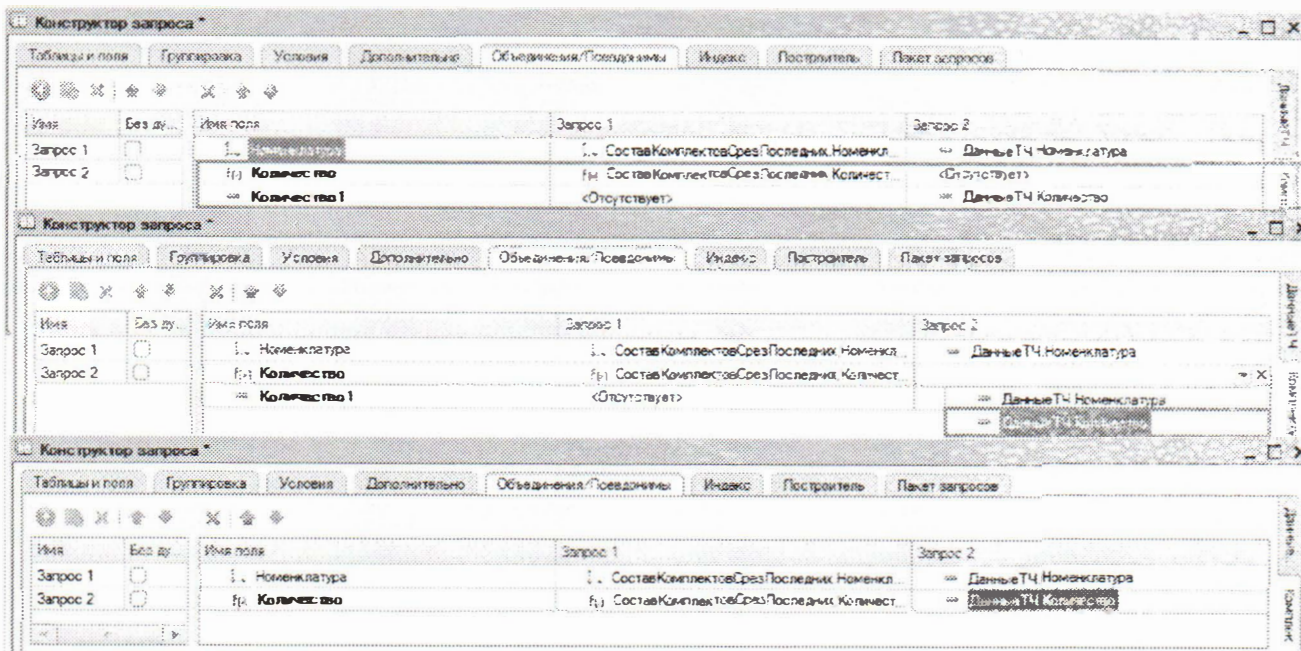


Рисунок 448

Добавим условие, что комплектом данная номенклатура не является (Рисунок 449).

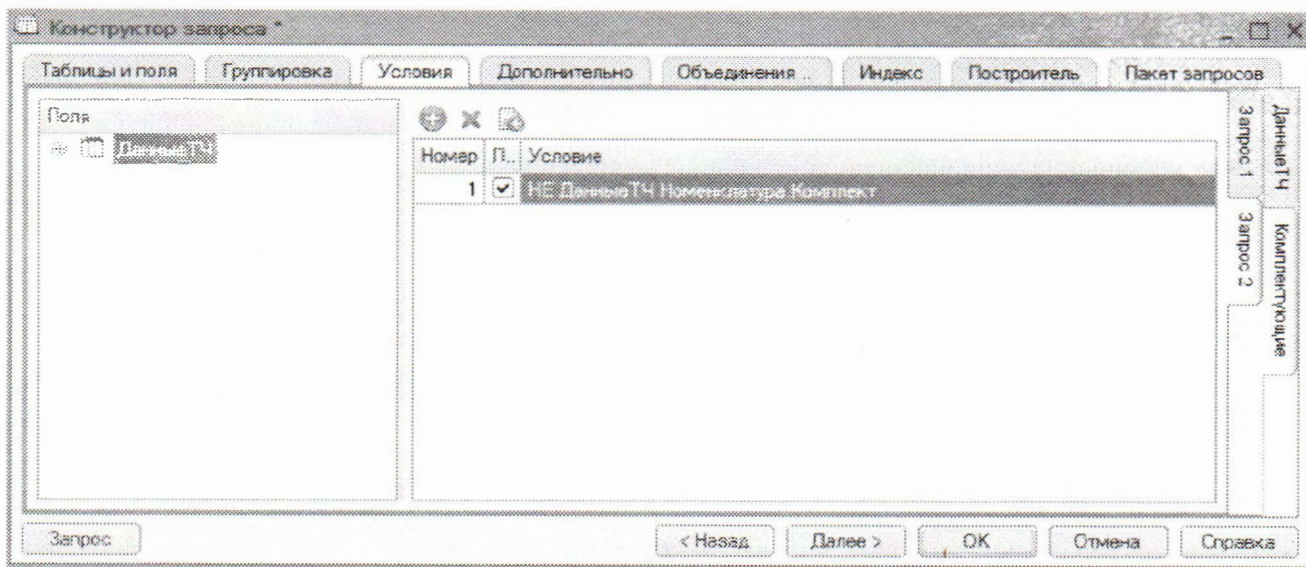


Рисунок 449

Таким образом, мы получили данные комплектующих и данные табличной части документа, которая не является комплектом. Далее для комплектующих следует получить

данные для корректного расчета себестоимости. Для этого добавим запрос пакета 3 и в качестве источника данных укажем таблицу остатков регистра бухгалтерии «Управленческий», из которого нужно получить данные для расчета себестоимости (Рисунок 450).

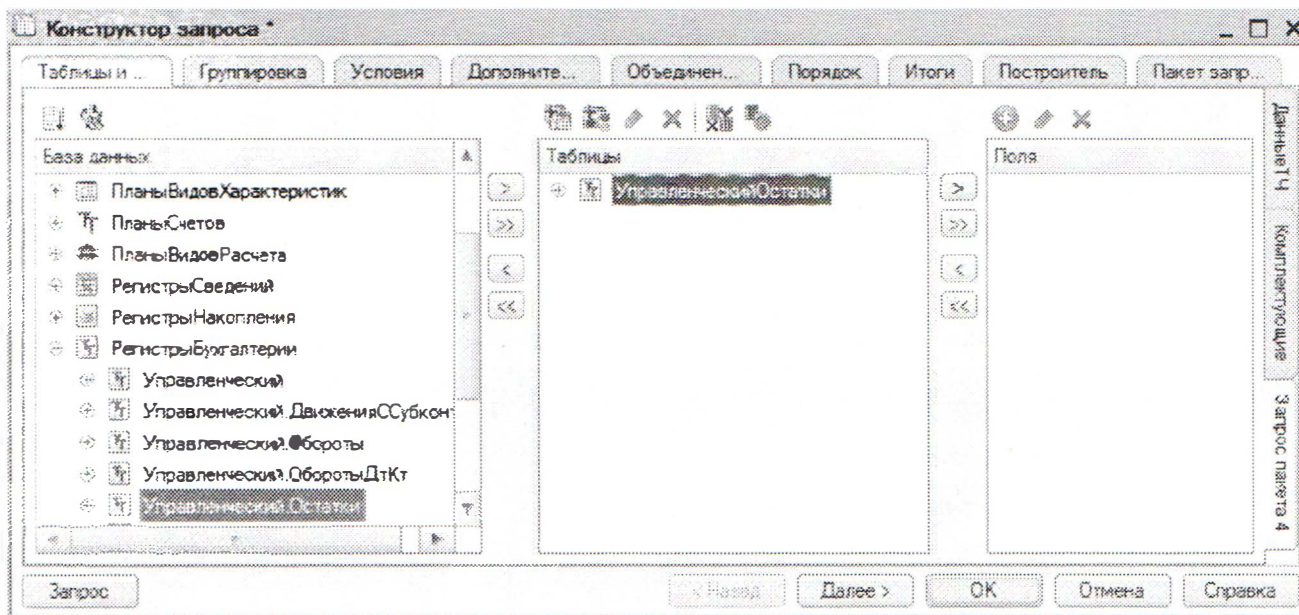


Рисунок 450

Обратим внимание на параметры данной таблицы (Рисунок 451).

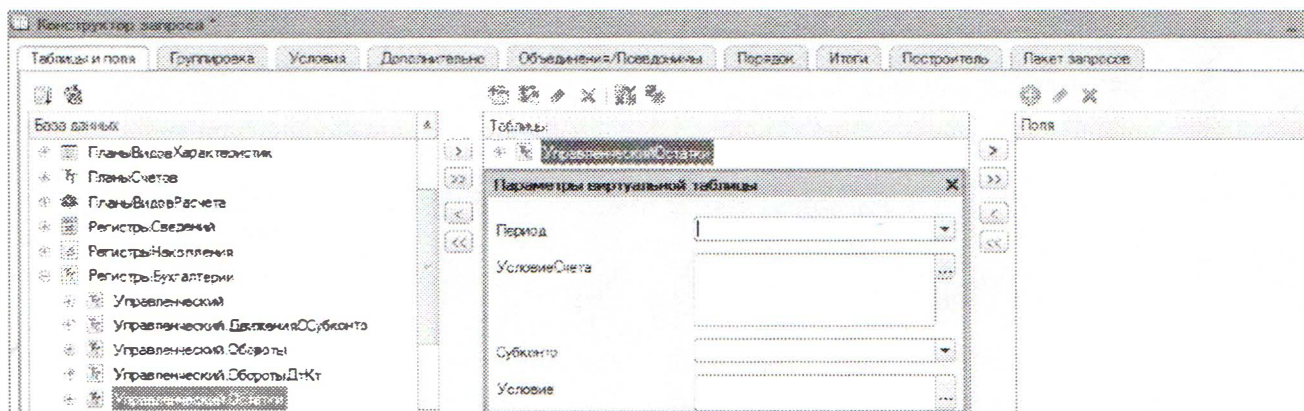


Рисунок 451

Заполним параметры виртуальной таблицы следующим образом (Рисунок 452):

- Период: &МоментВремени
- Условие счета: $Счет = ЗНАЧЕНИЕ(ПланСчетов.Управленческий.Товары)$

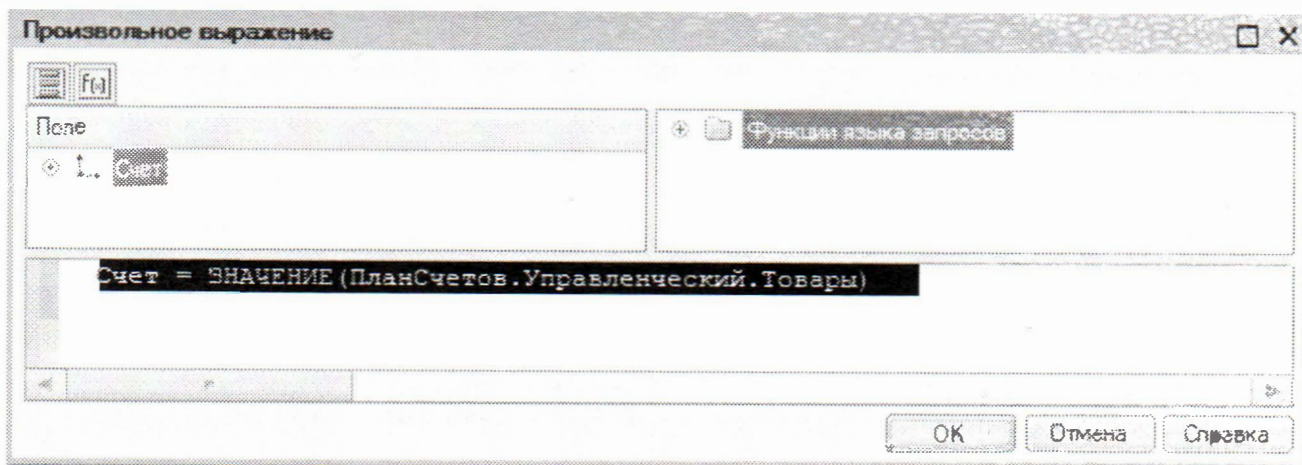


Рисунок 452

– Условие - поскольку аналитика ведется в разрезе номенклатуры и склада, то условие будет накладываться на субконто1 и субконто2 (Рисунок 453):

1. *Субконто1* – данные по номенклатуре;
2. *Субконто2* - &Склад.

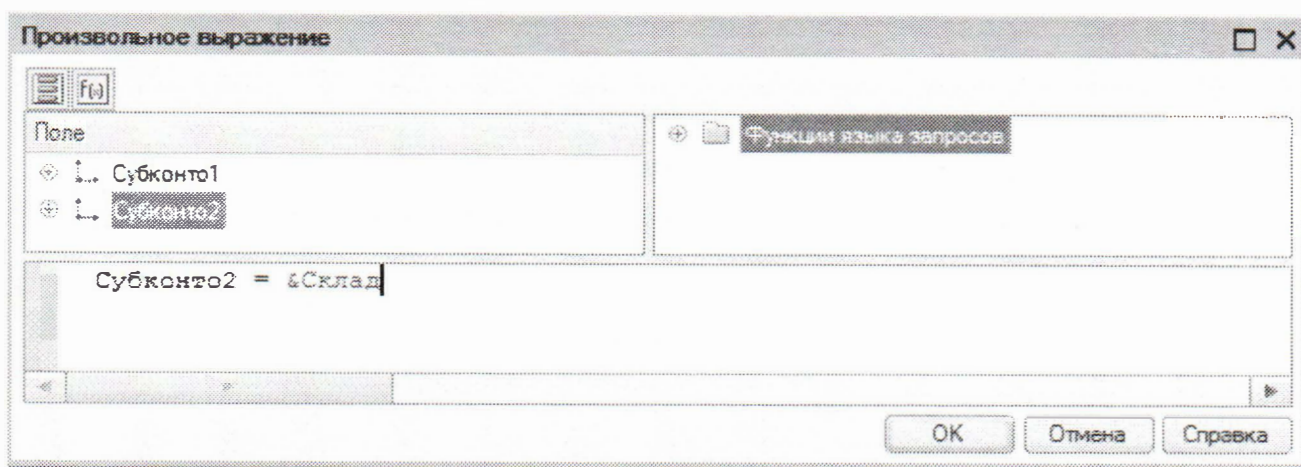


Рисунок 453

Получим данные по номенклатуре по Субконто1. Для этого воспользуемся тем же способом, что и ранее – с использованием временных таблиц в запросе пакета, и создадим новый запрос пакета, в котором нас будет интересовать поле «Номенклатура» из временной таблицы «Комплектующие» (Рисунок 454).

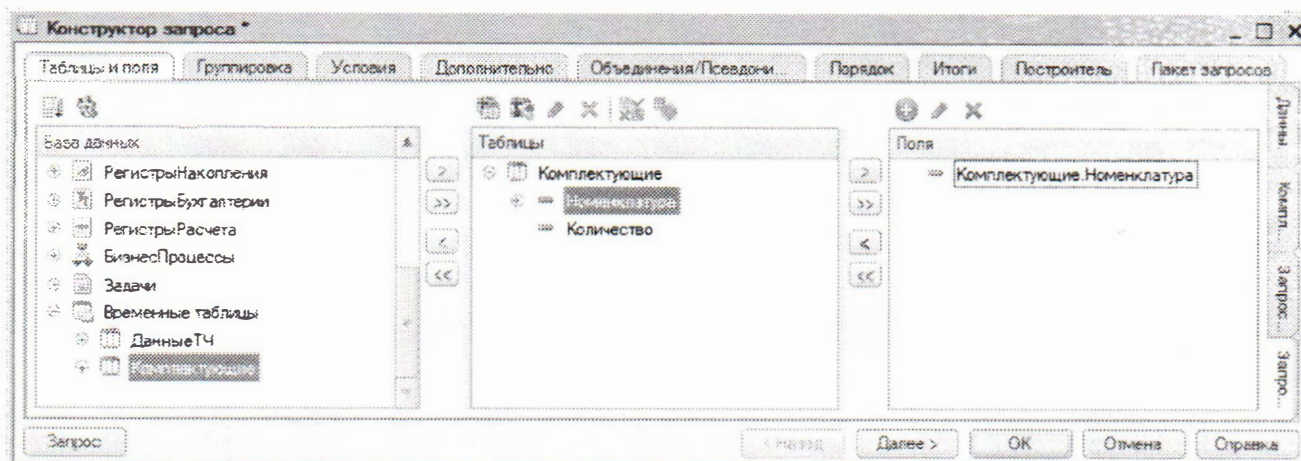


Рисунок 454

В тексте запроса скопируем фрагмент последнего пакета (Рисунок 455).

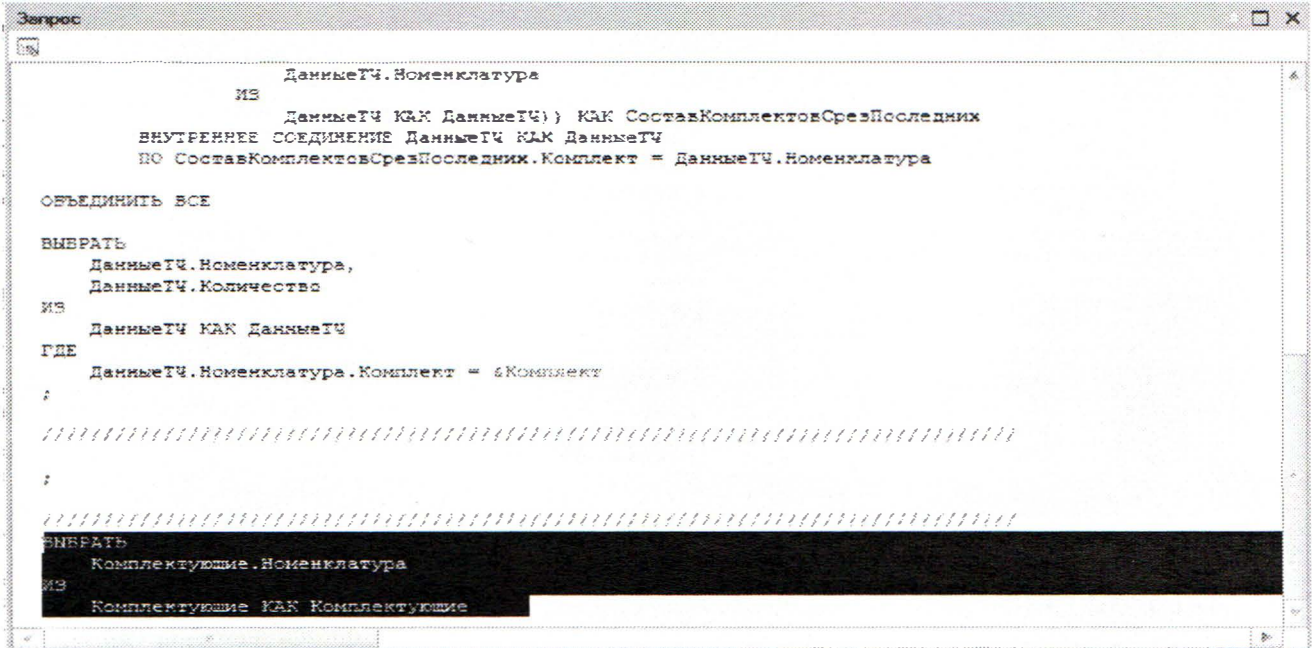


Рисунок 455

Вставим этот запрос в условие по Субконто1 и удалим последний запрос пакета (Рисунок 456).

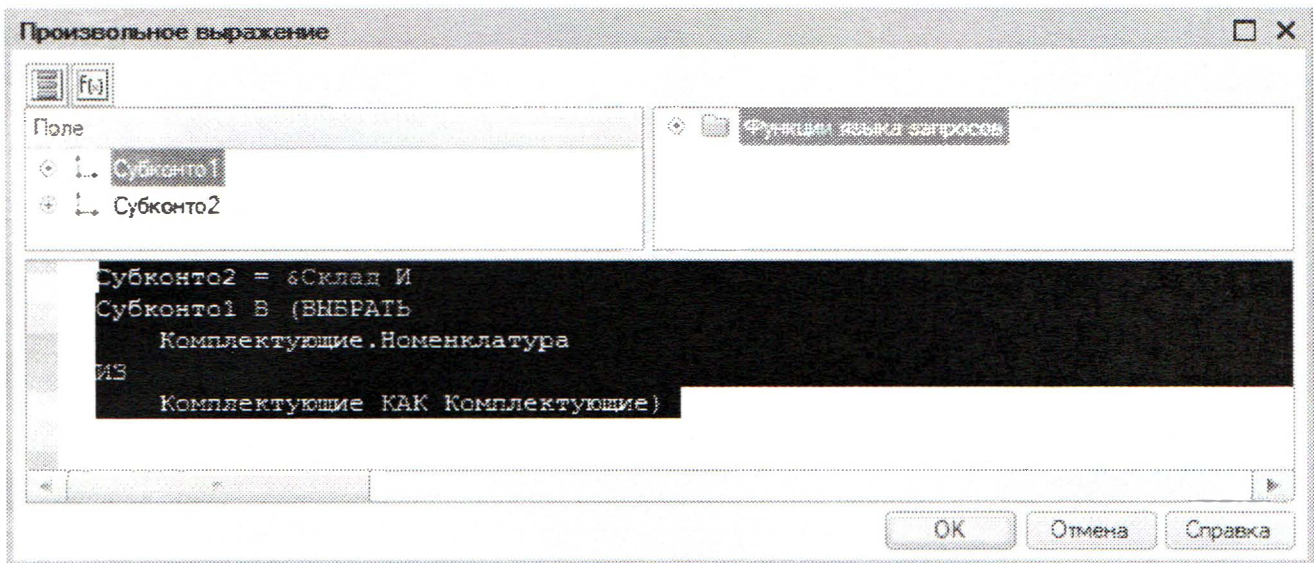


Рисунок 456

Когда все необходимые параметры виртуальной таблицы остатков бухгалтерского регистра заданы, для расчета себестоимости нам нужно взять данные по Субконто1 (номенклатура) и рассчитать себестоимость за каждую единицу: сумму по дебету разделить на количество по дебету (Рисунок 457).

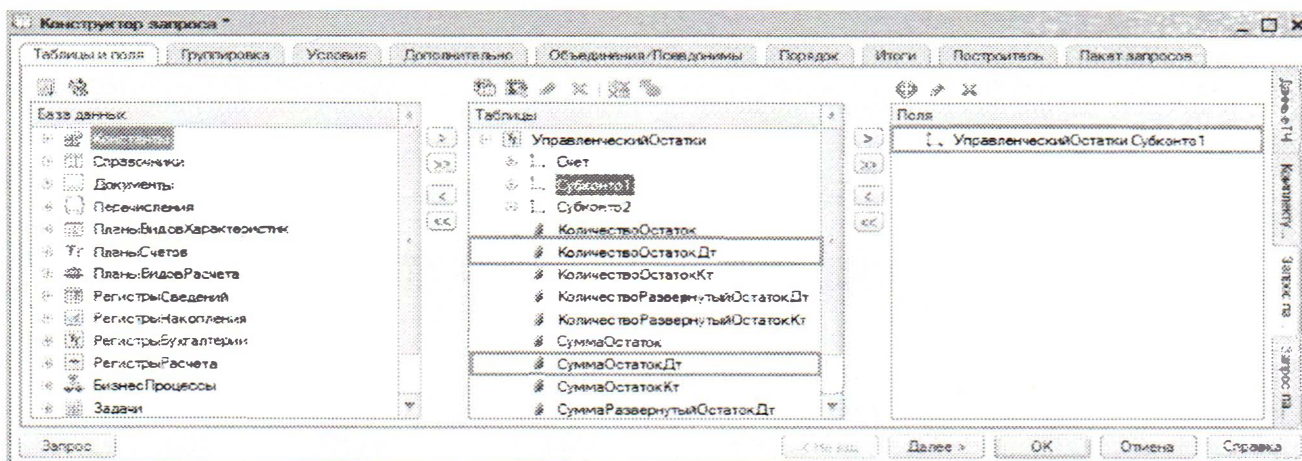


Рисунок 457

Переименуем поле «Субконто1» на «Номенклатура» на закладке «Объединения/псевдонимы» и приступим к расчету себестоимости (Рисунок 458).

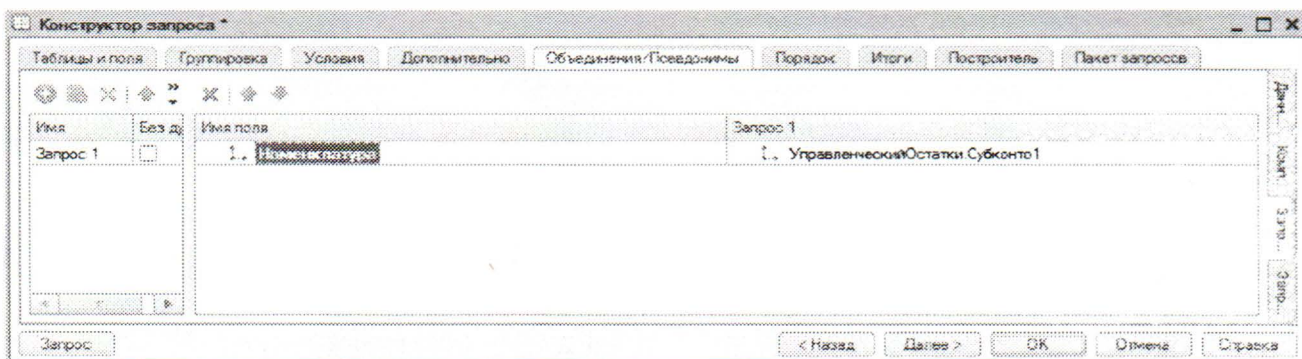


Рисунок 458

Как определили ранее, себестоимость единицы номенклатуры рассчитывается делением суммы по дебету на количество по дебету. Но значение поля «КоличествоОстатокДт» может быть как равным нулю, так и – в случае ошибок учета – иметь отрицательные значения. Для обработки таких ситуаций добавим новое рассчитываемое поле в запросе пакета и воспользуемся оператором выбора (Рисунок 459).

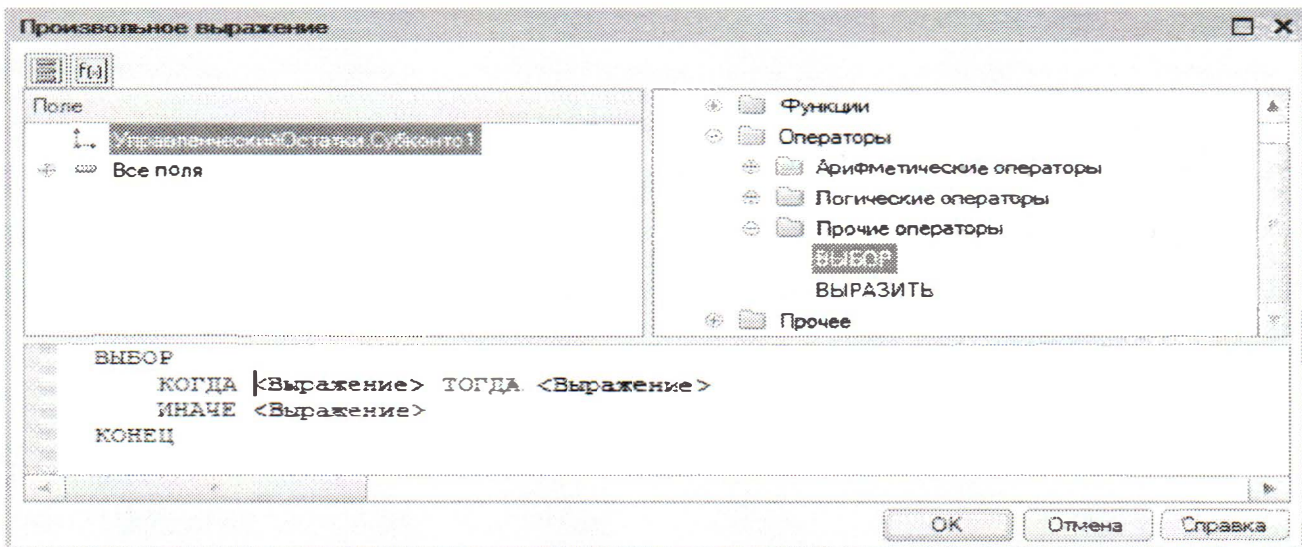


Рисунок 459

Зададим условие (Рисунок 460):

ВЫБОР

КОГДА *УправленческийОстатки.КоличествоОстатокДт* = 0 ТОГДА 0

ИНАЧЕ

УправленческийОстатки.СуммаОстатокДт/*УправленческийОстатки.КоличествоОстатокДт*

КОНЕЦ

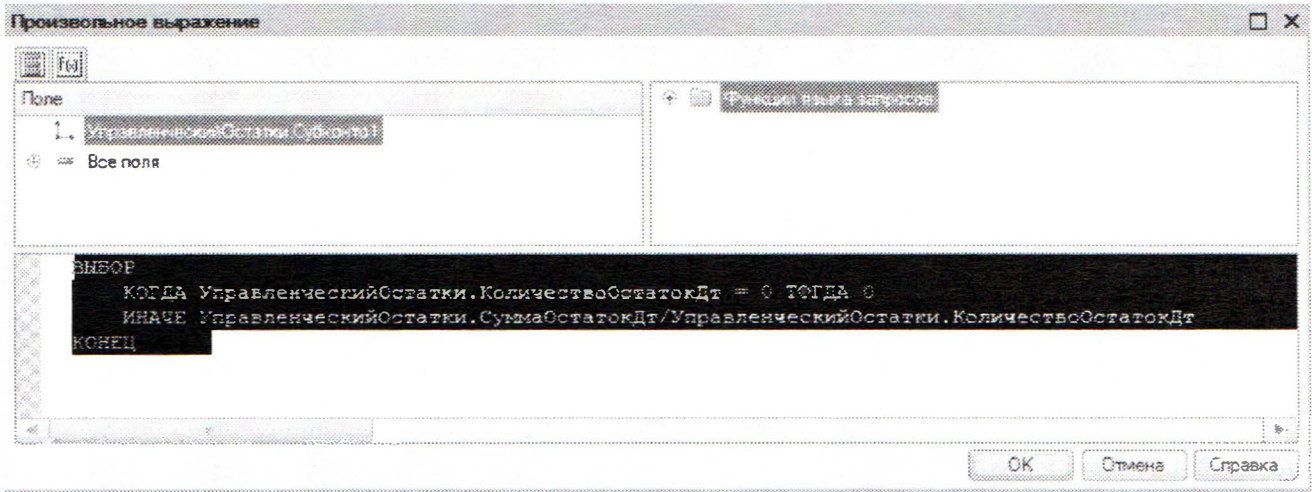


Рисунок 460

Переименуем это поле в «Себестоимость» на закладке «Объединения/псевдонимы» (Рисунок 461).

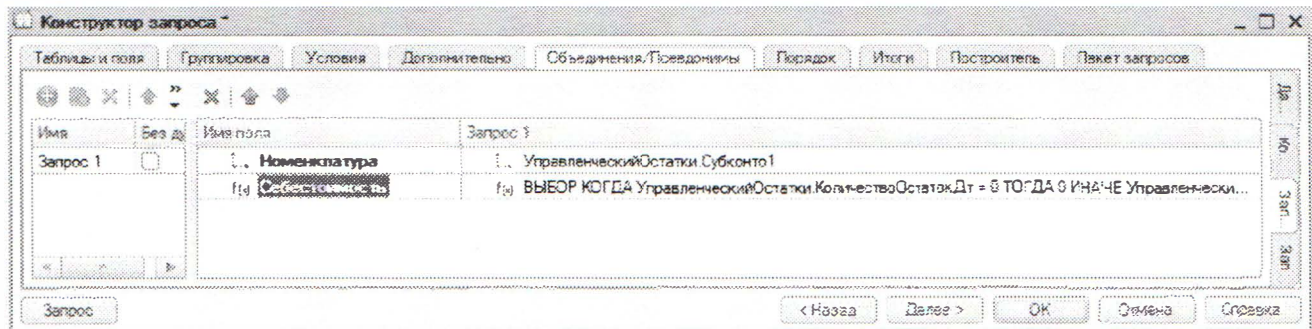


Рисунок 461

Следует обратить внимание на возможные значения результата формулы расчета себестоимости. Допустим, сумма по дебету будет равна 100, а количество по дебету – 33, тогда в результате расчета получим число с неограниченным количеством знаков после запятой. При округлении и последующем списании суммы по дебету путем умножения количества на себестоимость будут потеряны копейки. Для обработки этой ситуации можно связать таблицу «УправленческийОстатки» с временной таблицей «Комплектующие» в запросе пакета 3, но прежде, необходимо сгруппировать итоговое объединение таблиц «Комплектующие».

Для этого добавим еще один запрос пакета, в котором в качестве источника данных возьмем все поля из временной таблицы «Комплектующие» (Рисунок 462).

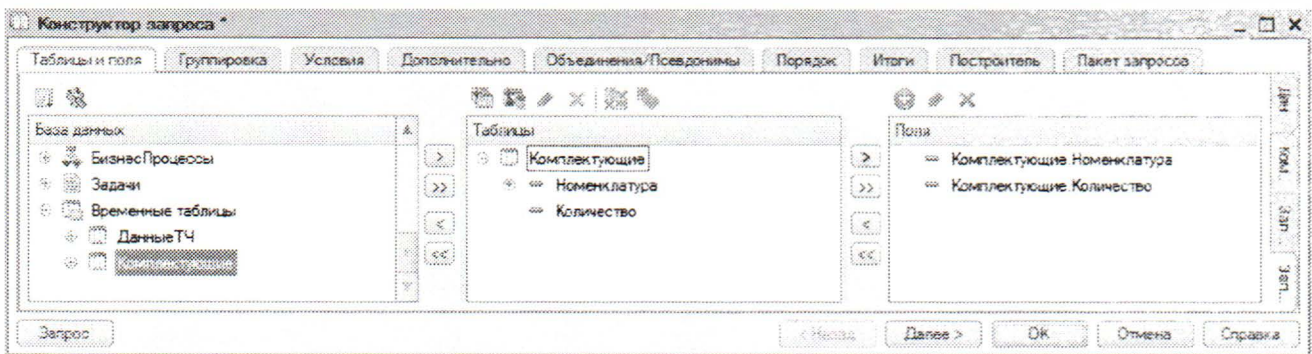


Рисунок 462

Для того чтобы записи по одной номенклатурной позиции не повторялись, и каждой из них соответствовала одна строка таблицы, на закладке «Группировка» сгруппируем поля по номенклатуре (Рисунок 463).

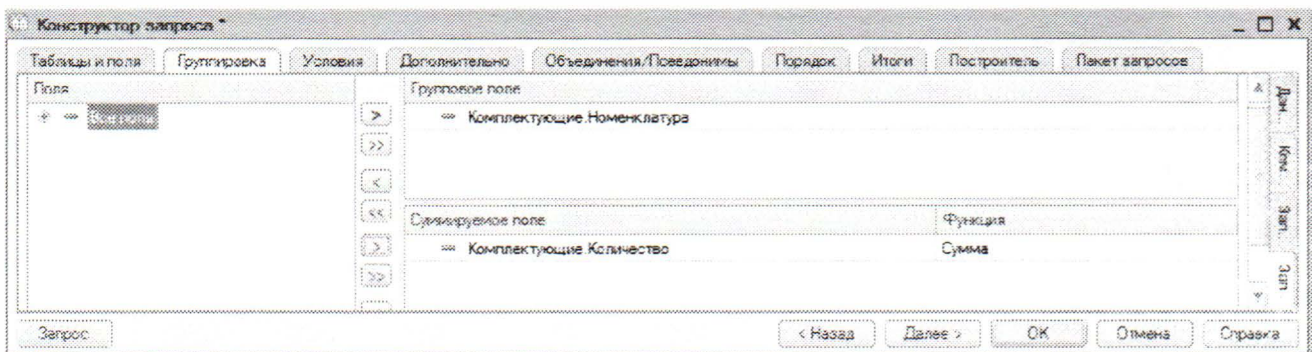


Рисунок 463

Поместим данные по сгруппированным комплектующим во временную таблицу «КомплГрупп» (Рисунок 464).

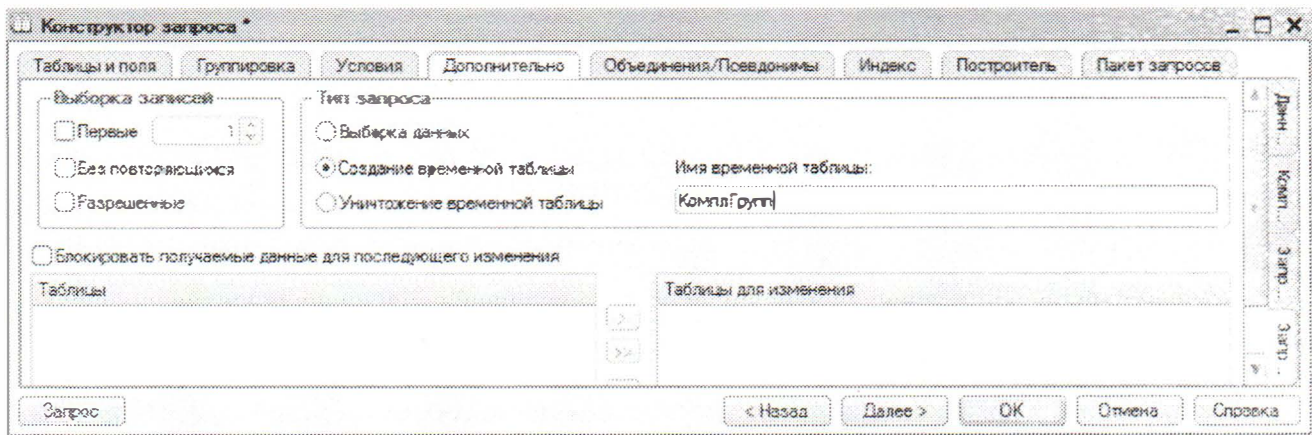


Рисунок 464

Далее используем эту таблицу в расчете себестоимости единицы номенклатуры: в случае если требуется списать все (количество по комплектующим сгруппированным совпадает с количеством по остаткам дебета), списываем всю сумму по дебету, иначе – рассчитываем от деления суммы на количество остатка и умножаем на количество комплектов.

Для этого на закладке «Пакеты запросов» перенесем пакет «КомплГрупп» на позицию выше запроса, в котором рассчитываем себестоимость для того чтобы была возможность к

нему обращаться. Далее выберем временную таблицу «КомплГрупп» в качестве источника данных в запросе пакета расчета себестоимости (Рисунок 465).

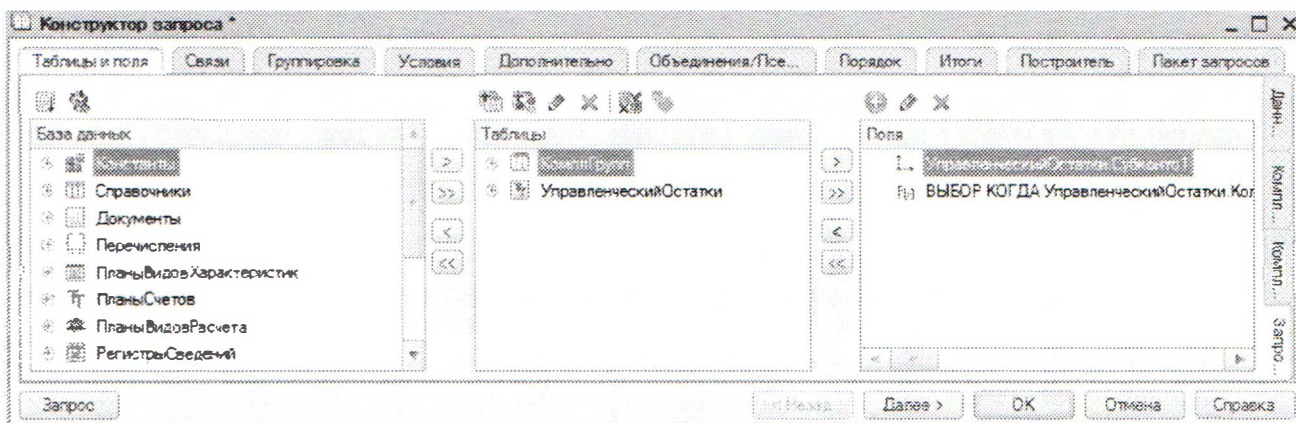


Рисунок 465

Изменим условие рассчитываемого поля себестоимости (Рисунок 466):

ВЫБОР

КОГДА УправленческийОстатки.КоличествоОстатокДт = 0

ТОГДА 0

КОГДА КомплГрупп.Количество=УправленческийОстатки.КоличествоОстатокДт

ТОГДА УправленческийОстатки.СуммаОстатокДт

ИНАЧЕ (УправленческийОстатки.СуммаОстатокДт /

*УправленческийОстатки.КоличествоОстатокДт)*КомплГрупп.Количество*

КОНЕЦ

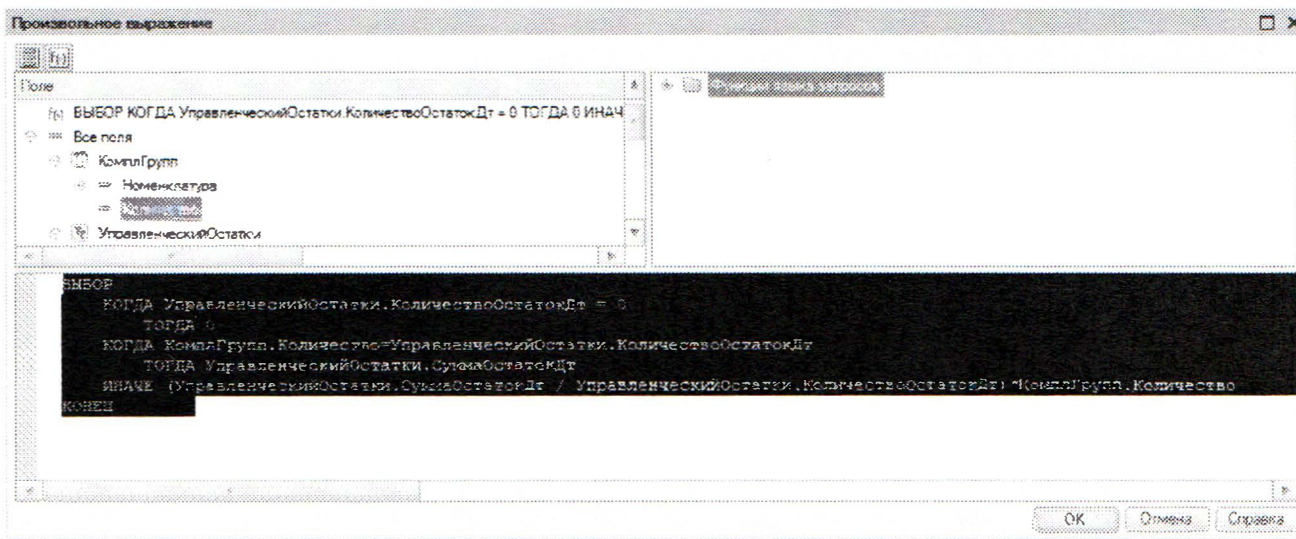


Рисунок 466

Если мы перенесем «*КомплГрупп.Количество» до знака деления «/», то это поможет избежать в дальнейшем ошибок округления (Рисунок 467).

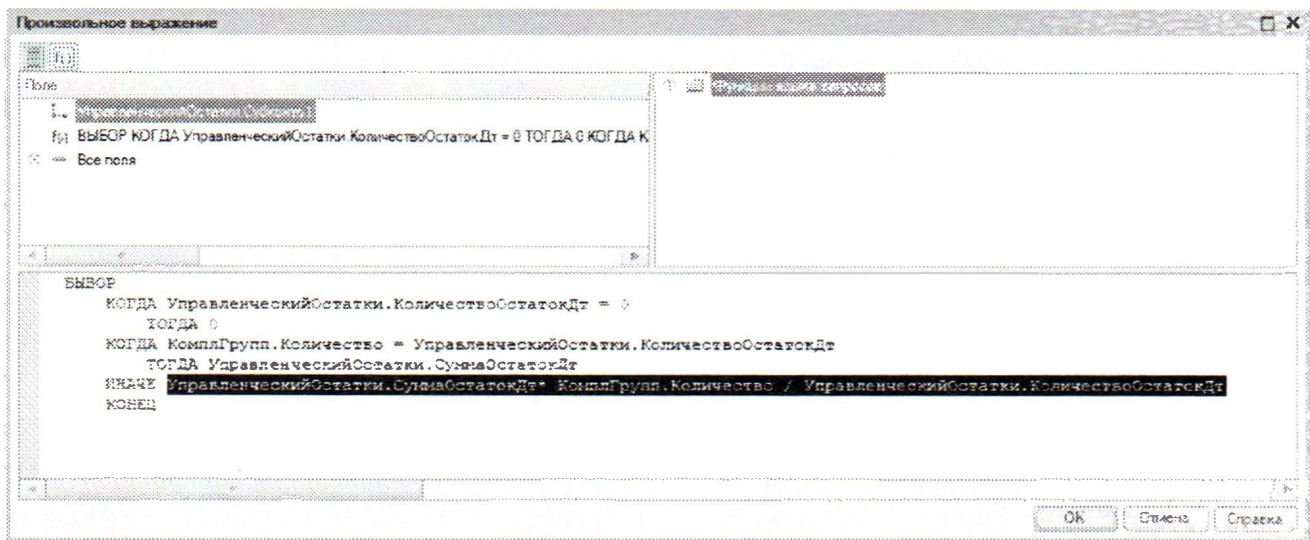


Рисунок 467

В случае, если списываем сумму по всем количествам комплектов, то ошибки округления исключены и тогда строка условие:

$$\begin{aligned}
 & \text{КОГДА} && \text{КомплГрупп.Количество} && = \\
 & \text{УправленческийОстатки.КоличествоОстатокДт} && && \\
 & \text{ТОГДА} && \text{УправленческийОстатки.СуммаОстатокДт} &&
 \end{aligned}$$

– уже не имеет смысла.

Удалим ее, получив конечную формулу расчета себестоимости, и перейдем к связыванию таблиц (Рисунок 468).

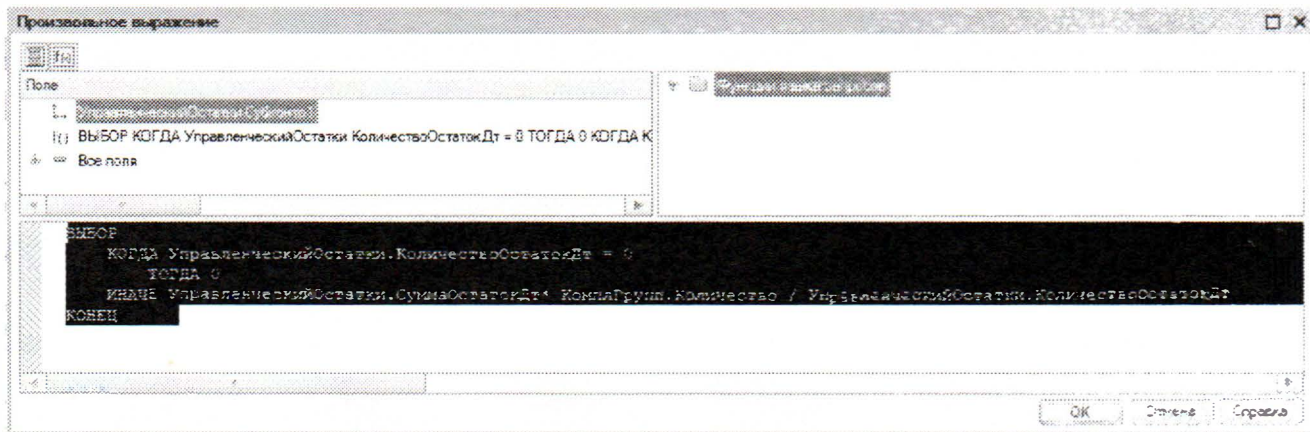


Рисунок 468

Для связи таблиц возьмем все, что есть из сгруппированных комплектов и найдем соответствия по номенклатуре в управленческом регистре бухгалтерии. Получим условие связи (Рисунок 469).

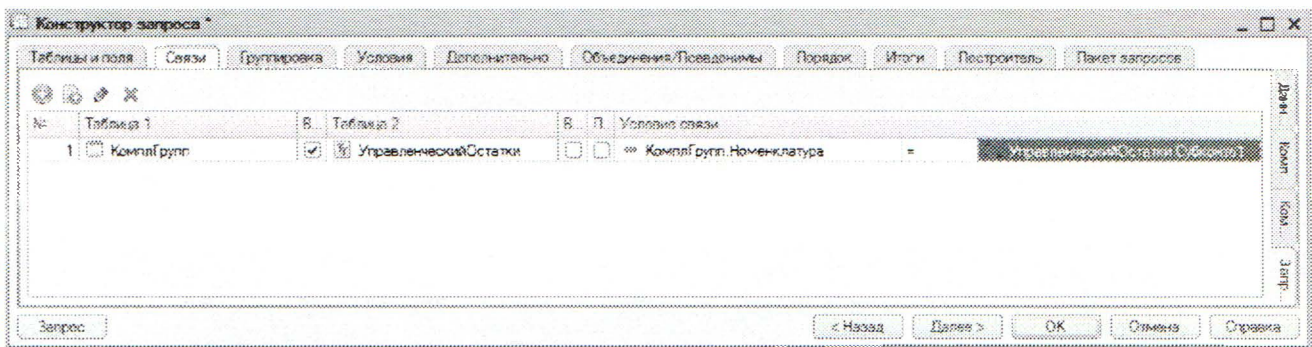


Рисунок 469

Добавим поля сгруппированных комплектов к таблице запроса пакета 3 (Рисунок 470).

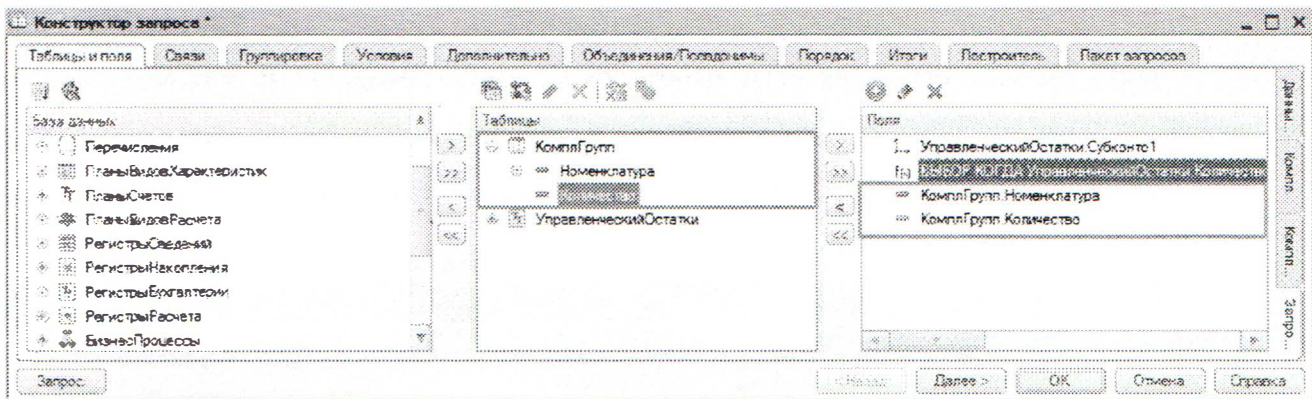


Рисунок 470

При этом поле `УправленческийОстатки.Субконто1` можно удалить, так как значение номенклатуры теперь получаем из `КомплГрупп.Номенклатура`. Далее переименуем его на закладке псевдонимов из «`Номенклатура1`» в «`Номенклатура`» (Рисунок 471).

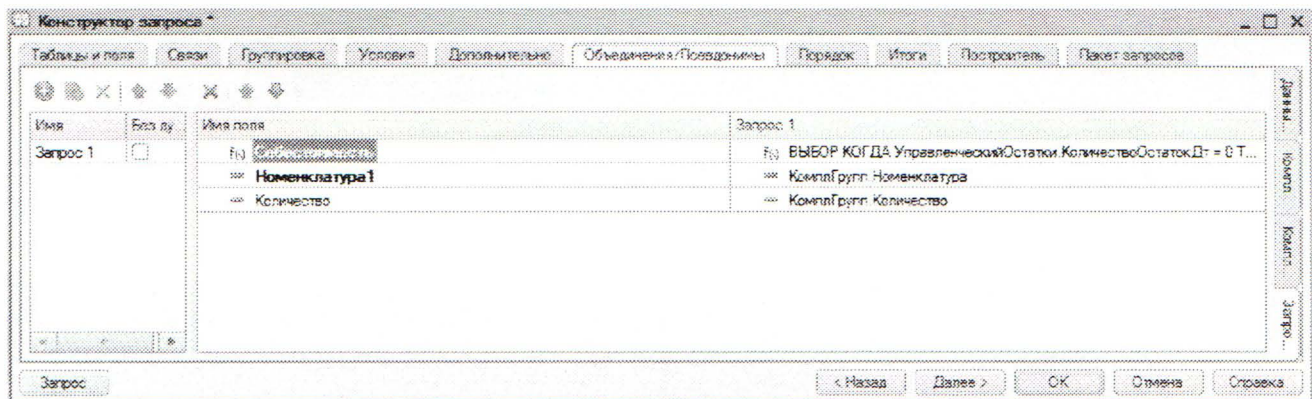


Рисунок 471

Теперь основная сложность в расчетах себестоимости состоит в том, что поля: *УправленческийОстатки.КоличествоОстатокДт* и *УправленческийОстатки.СуммаОстатокДт* могут быть равны NULL. Для обработки такой ситуации, применительно ко всем количественным полям условия расчета себестоимости, используем «ЕСТЬNULL» (Рисунок 472).

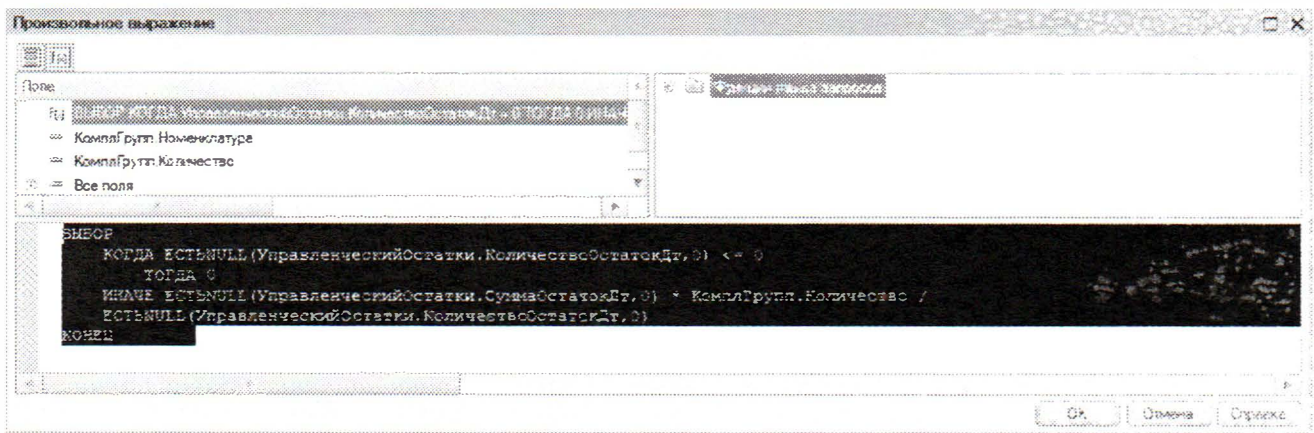


Рисунок 472

При расчете себестоимости необходимо делать проверку на наличие остатков на складе – хватает ли их для списания, в противном случае – выводить сообщение пользователю. Для этого добавим в Запрос пакета 3 еще одно рассчитываемое поле – «КолОстаток» (Рисунок 473).

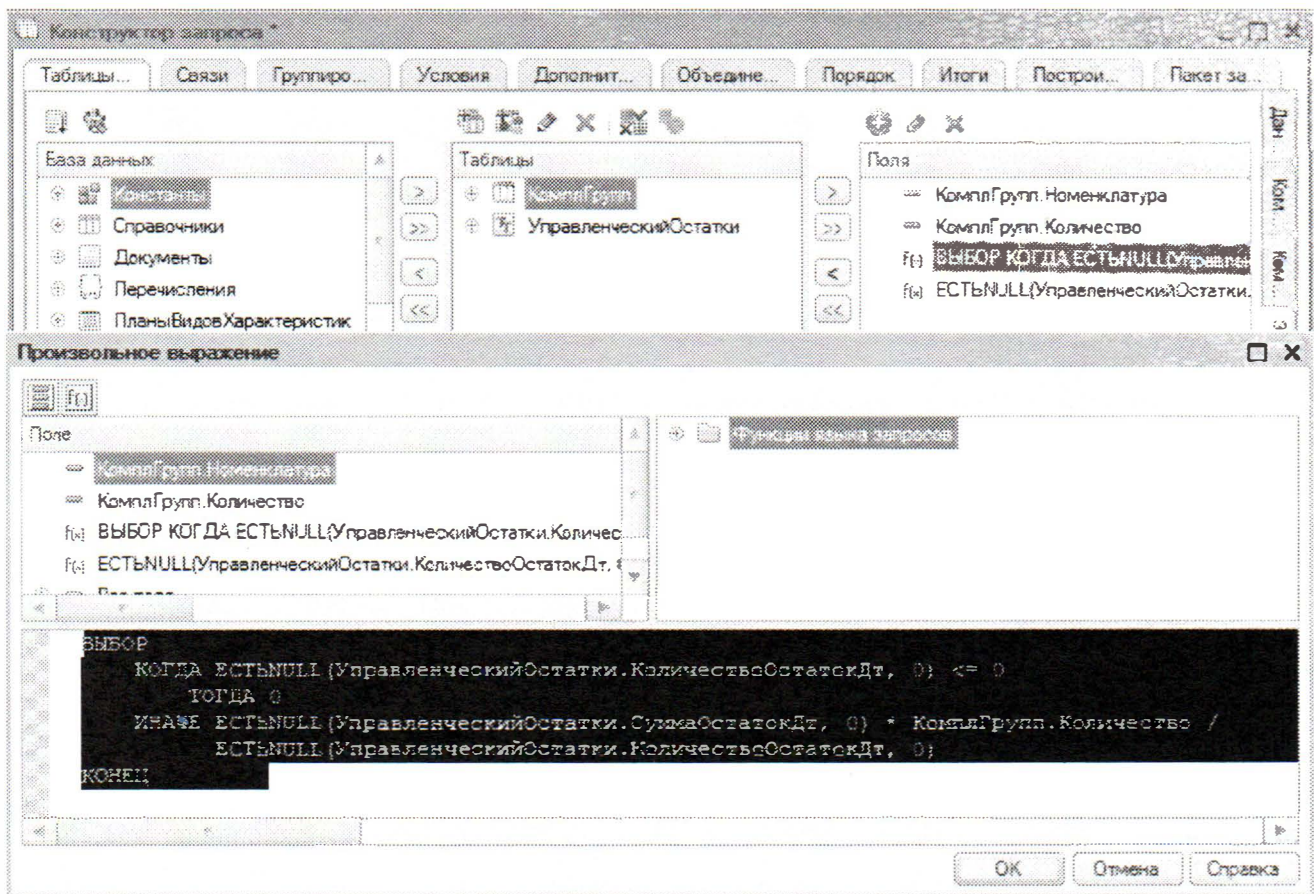


Рисунок 473

Итак, у нас есть данные по комплектующим с их себестоимостью. Теперь нам нужен запрос, который бы получал данные из табличной части по комплектам. Использовать для этих целей временную таблицу «ДанныеТЧ» мы не можем, так как там будет одна строка с общим количеством, что нам не подходит, поэтому создадим еще один запрос пакета и получим данные из табличной части «ДанныеТЧ» (Рисунок 474).

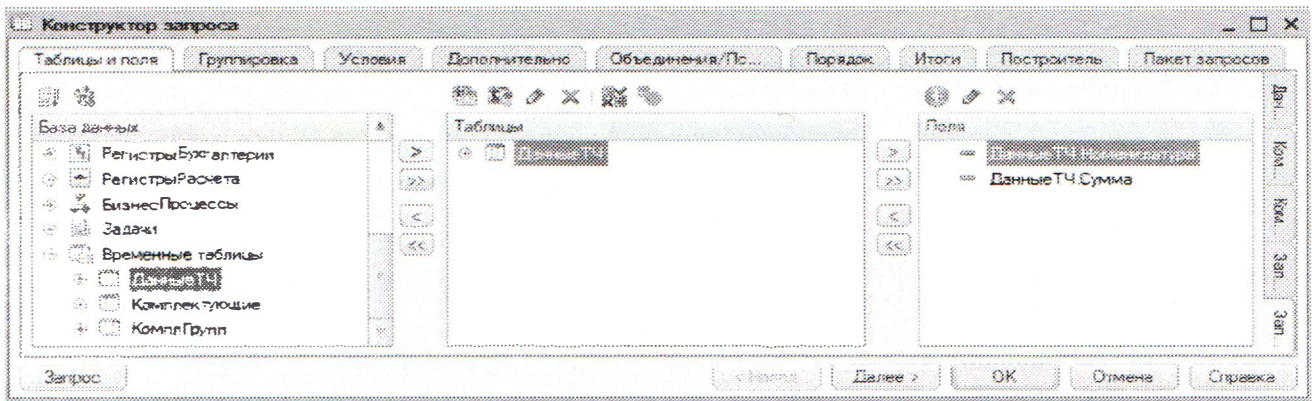


Рисунок 474

Итоговый текст запроса выглядит следующим образом:

"ВЫБРАТЬ

РасходнаяНакладнаяСписокНоменклатуры.Номенклатура КАК Номенклатура,
 СУММА(РасходнаяНакладнаяСписокНоменклатуры.Количество) КАК Количество,
 СУММА(РасходнаяНакладнаяСписокНоменклатуры.Сумма) КАК Сумма

ПОМЕСТИТЬ ДанныеТЧ

ИЗ

Документ.РасходнаяНакладная.СписокНоменклатуры КАК

РасходнаяНакладнаяСписокНоменклатуры

ГДЕ

РасходнаяНакладнаяСписокНоменклатуры.Ссылка = &Ссылка

(ГРУППИРОВАТЬ ПО

РасходнаяНакладнаяСписокНоменклатуры.Номенклатура

ИНДЕКСИРОВАТЬ ПО

Номенклатура

;

////////////////////////////////////

ВЫБРАТЬ

СоставКомплектовСрезПоследних.Номенклатура,
 СоставКомплектовСрезПоследних.Количество * ДанныеТЧ.Количество КАК

Количество

ПОМЕСТИТЬ Комплектующие

ИЗ

РегистрСведений.СоставКомплектов.СрезПоследних(

&МоментВремени,

Номенклатура В

(ВЫБРАТЬ

ДанныеТЧ.Номенклатура

ИЗ

СоставКомплектовСрезПоследних

ВНУТРЕННЕЕ СОЕДИНЕНИЕ ДанныеТЧ КАК ДанныеТЧ

ПО СоставКомплектовСрезПоследних.Комплект = ДанныеТЧ.Номенклатура

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

ДанныеТЧ.Номенклатура,

ДанныеТЧ.Количество

ИЗ

ДанныеТЧ КАК ДанныеТЧ

ГДЕ

НЕ ДанныеТЧ.Номенклатура.Комплект

;

ВЫБРАТЬ

Комплектующие.Номенклатура,

СУММА(Комплектующие.Количество) КАК Количество

ПОМЕСТИТЬ КомплГрупп

ИЗ

Комплектующие КАК Комплектующие

СГРУППИРОВАТЬ ПО

Комплектующие.Номенклатура

;

ВЫБРАТЬ

КомплГрупп.Номенклатура КАК Номенклатура,

КомплГрупп.Количество,

ВЫБОР

КОГДА «ЕСТЬNULL»(УправленческийОстатки.КоличествоОстатокДт, 0)

<= 0

ТОГДА 0

ИНАЧЕ «ЕСТЬNULL»(УправленческийОстатки.СуммаОстатокДт, 0) *

КомплГрупп.Количество / «ЕСТЬNULL»(УправленческийОстатки.КоличествоОстатокДт, 0)

КОНЕЦ КАК Себестоимость,

«ЕСТЬNULL»(УправленческийОстатки.КоличествоОстатокДт, 0) КАК КолОстаток

ИЗ

КомплГрупп КАК КомплГрупп

ЛЕВОЕ СОЕДИНЕНИЕ РегистрБухгалтерии.Управленческий.Остатки(

&МоментВремени,
 Счет = ЗНАЧЕНИЕ(ПланСчетов.Управленческий.Товары),
 ,
 Субконто2 = &Склад
 И Субконто1 В
 (ВЫБРАТЬ
 Комплектующие.Номенклатура
 ИЗ
 Комплектующие КАК

Комплектующие)) КАК Управленческий(Остатки

ПО Компл.Групп.Номенклатура = Управленческий(Остатки.Субконто1

////////////////////////////////////
 ВЫБРАТЬ

ДанныеТЧ.Номенклатура,

ДанныеТЧ.Сумма

ИЗ

ДанныеТЧ КАК ДанныеТЧ";

5.2.2 Бухгалтерский учет. Таблицы.

Рассмотрим регистры бухгалтерии и их свойства.

Для этого запустим приложение в режиме «1С:Предприятие» и откроем консоль запросов. Нас интересуют виртуальные таблицы регистра бухгалтерии «Управленческий» (Рисунок 475).

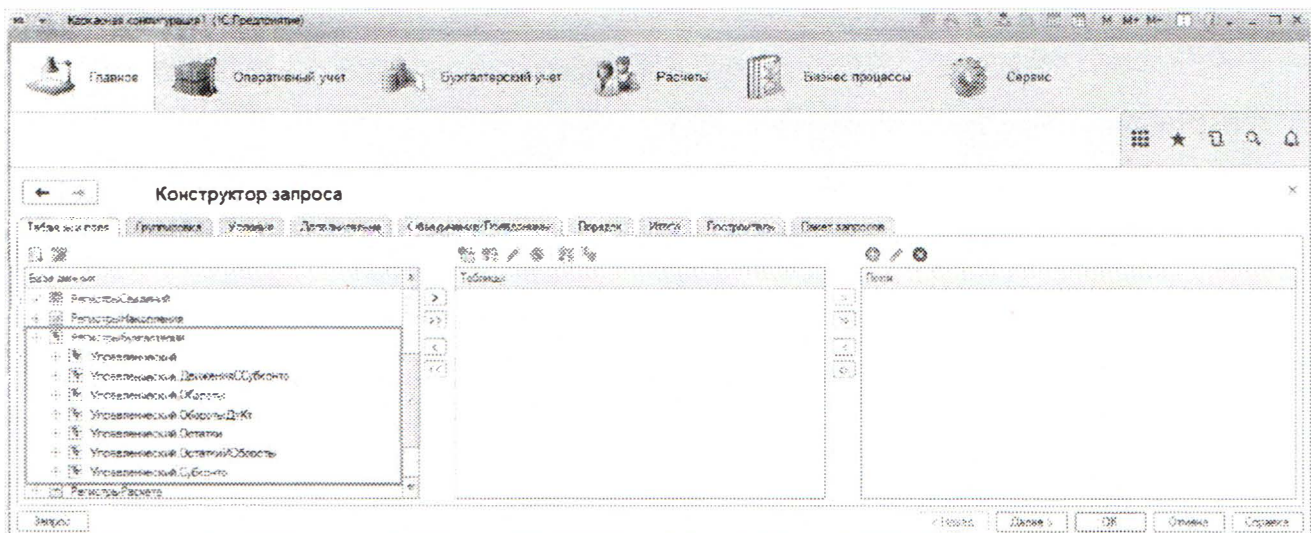


Рисунок 475

Рассмотрим параметры виртуальной таблицы остатков регистра «Управленческий». Цель работы с таблицей – получить остатки по счету в разрезе какой-либо аналитики (Рисунок 476).

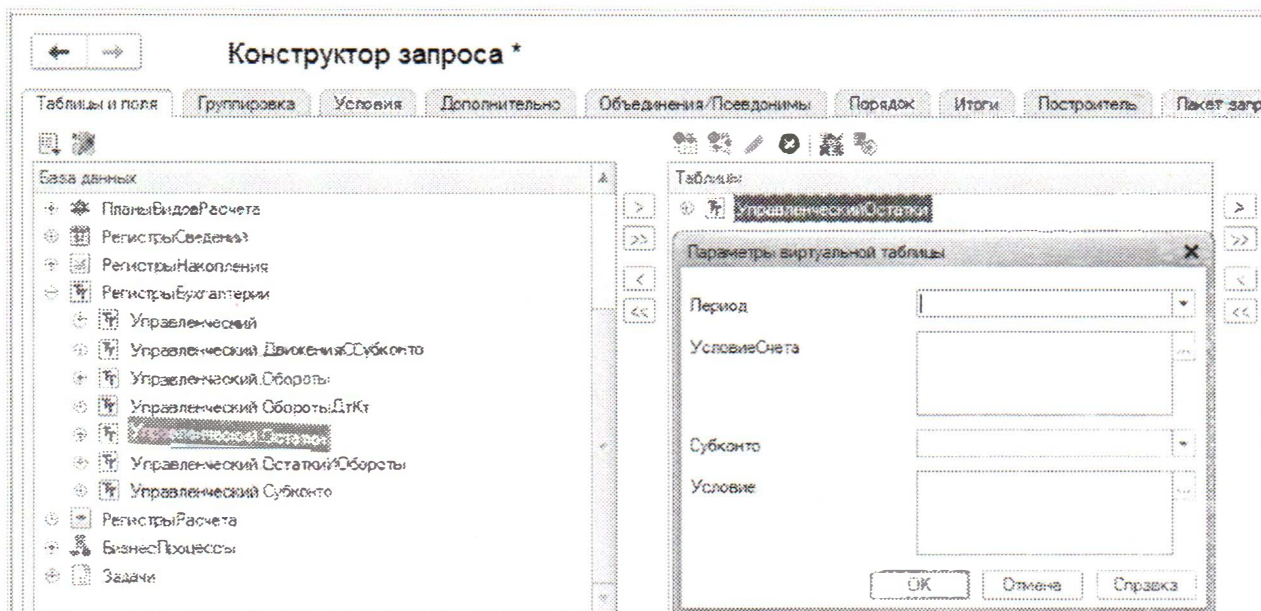


Рисунок 476

- Период – параметр, который определяет, на какую дату требуется получить остатки по счету;
- (&МоментВремени);
- Условие счета – параметр, который определяет, по какому счету требуется получить остатки;
- (Счет = ЗНАЧЕНИЕ(ПланСчетов.Управленческий.Товары));
- Субконто – параметр, который определяет состав и порядок следования субконто в запросе (может принимать на вход как список значений, так и массив);
- Условие – параметр, который определяет условие, накладываемое на субконто (в данном случае: номенклатура и склад);

Рассмотрим на практике использование параметра «Субконто». Для этого выберем все поля таблицы и заполним параметры запроса (Рисунок 477).

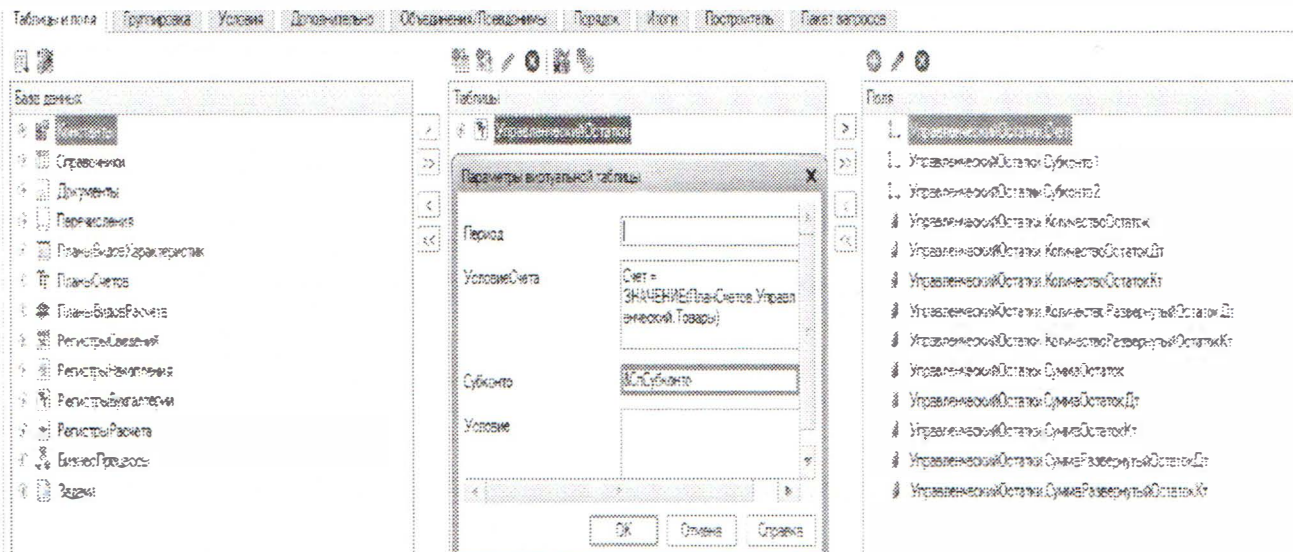


Рисунок 477

На закладке «Параметры» консоли запроса добавим параметр «СпСубконто» со значением «СписокЗначений» и заполним список субконто predetermined types of subaccounts: «Номенклатура» и «Склад» (Рисунок 478).

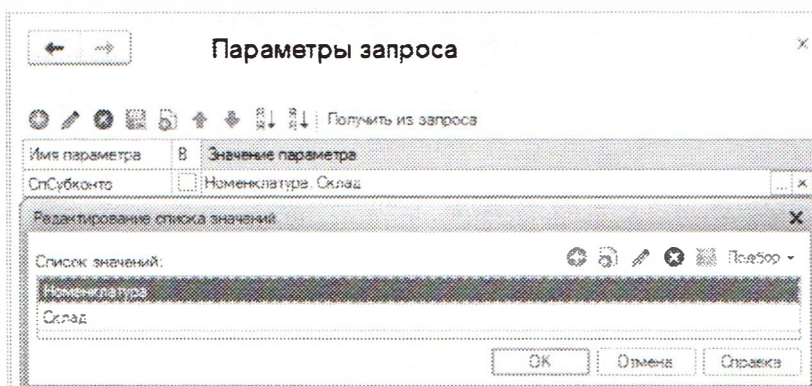


Рисунок 478

Результат выполнения запроса состоит из 2-х номенклатурных позиций. Следует обратить внимание на порядок очередности колонок: Субконто1 (номенклатура)→ Субконто2 (склад) – именно в том порядке, который мы указали в списке значений параметра «СпСубконто» (Рисунок 479).

Счет	Субконто1	Субконто2	КоличествоОстаток	КоличествоОстатокДт	КоличествоОстатокКт
Товары	Вещи KGS 3760 IE	Основной	2,000	2,000	
Товары	Вещи ER 9007 E	Основной	1,000	1,000	

Рисунок 479

Поменяем местами очередность следования субконто в списке значений параметра и выполним запрос (Рисунок 480).

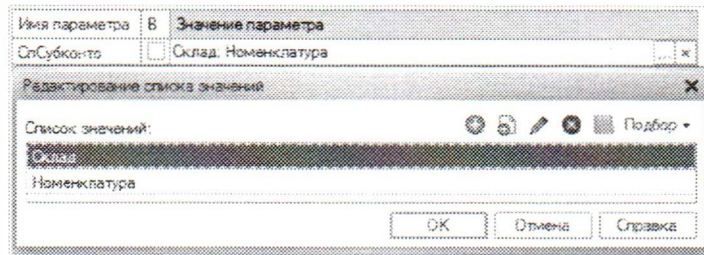


Рисунок 480

В таблице результата запроса порядок колонок в запросе поменялся местами, согласно очередности в списке значений параметра (Рисунок 481).

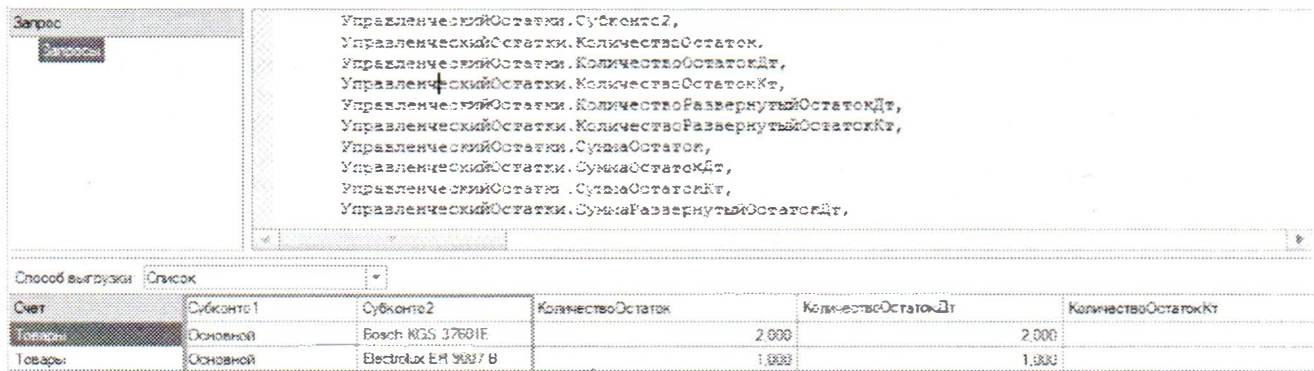


Рисунок 481

Удалим из списка значений субконто «Склад» и получим результат запроса (Рисунок 482).

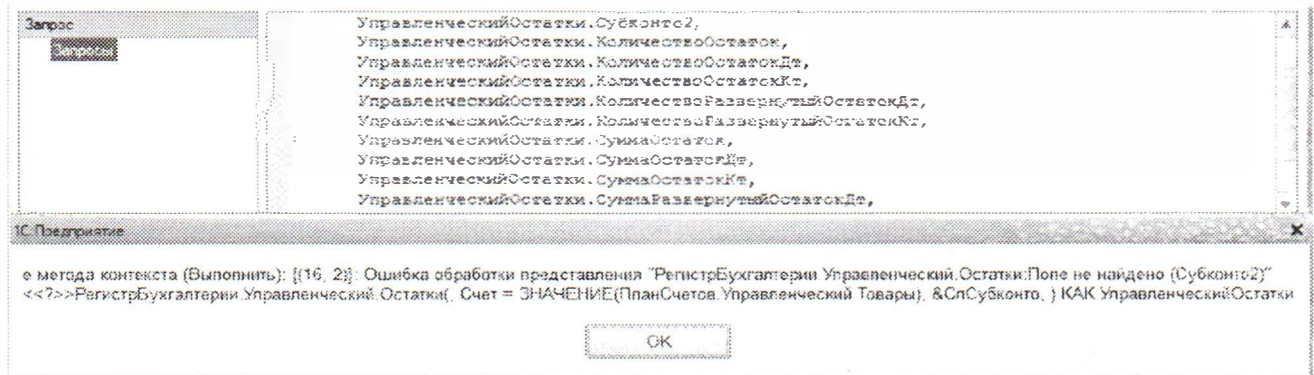


Рисунок 482

Ошибка связана с тем, что в полях запроса указаны как субконто1, так и субконто2 (Рисунок 483).

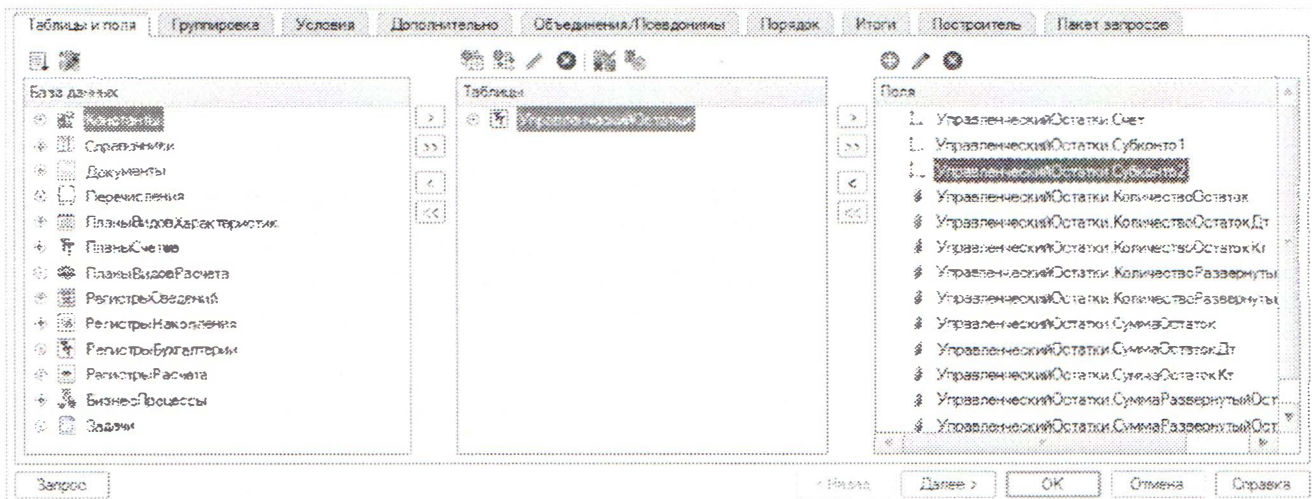


Рисунок 483

Удалим поле *УправленческийОстатки.Субконто2* из запроса и выполним его повторно (Рисунок 484).

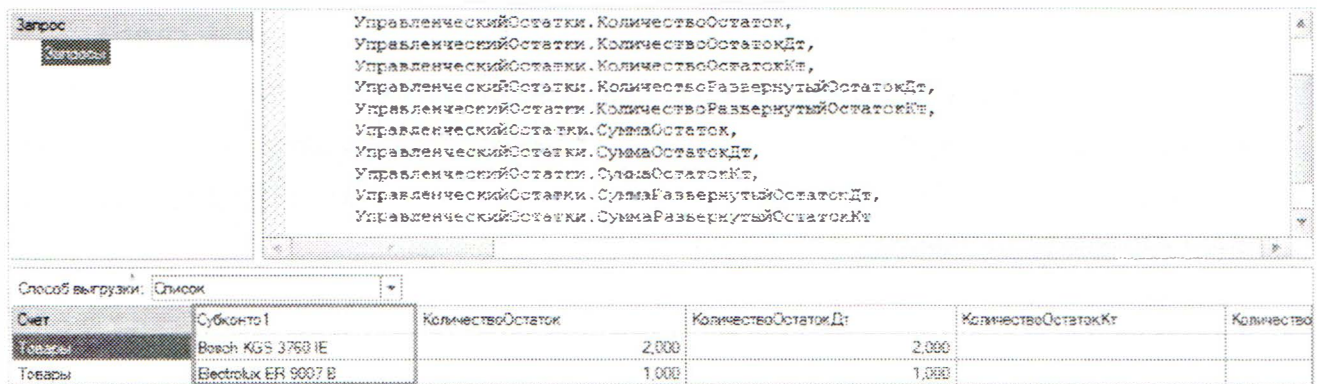


Рисунок 484

В таблице результата запроса увидим данные только по Субконто1 – «Номенклатура», как и было указано в параметрах. Таким образом, несмотря на то, какие субконто, и в каком порядке они указаны в запросе, с помощью параметра виртуальной таблицы «Субконто» можно управлять ими. Это особенно полезно, когда требуется обработать данные по большому количеству счетов сразу и, например, в случае, когда одна и та же аналитика на разных счетах является разными видами субконто, но при этом нужно получить единые данные. Для решения таких задач можно передать в параметр «Субконто» и список, и порядок вывода нужных субконто (кроме ситуаций, когда нужного субконто на счете нет).

Рассмотрим параметры виртуальной таблицы оборотов регистра «Управленческий» (Рисунок 485):

- НачалоПериода;
- КонецПериода;
- Периодичность;
- УсловиеСчета;
- Субконто;

- Условие;
- УсловиеКорСчета;
- КорСубконто.

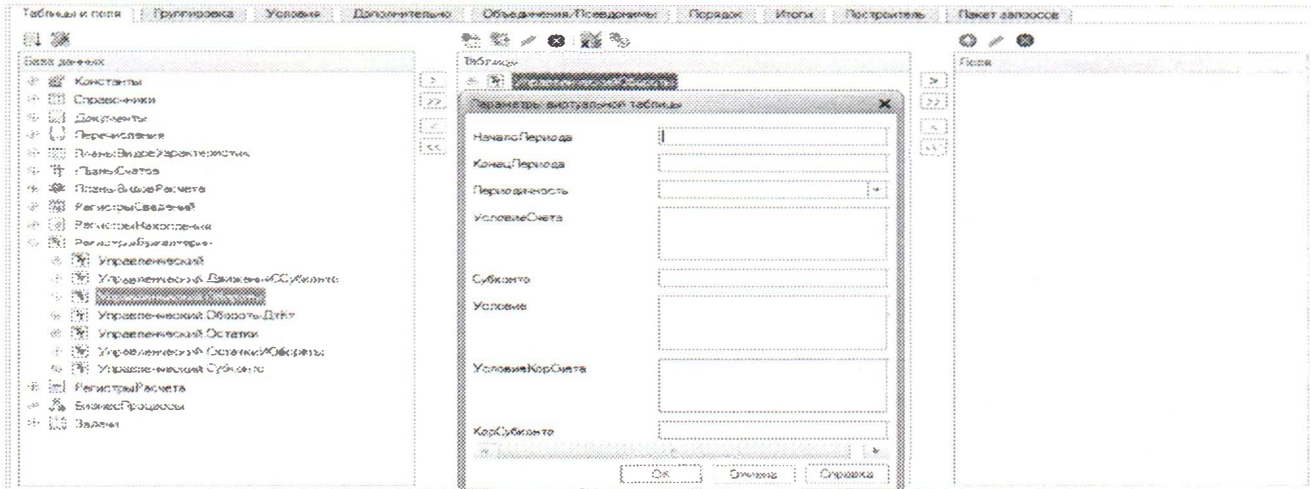


Рисунок 485

Некоторые из этих параметров были рассмотрены ранее, кроме *УсловиеКорСчета* и *КорСубконто*. Обратим внимание на их свойства. Если мы попытаемся задать общий параметр «Условие», то увидим, что для него доступны поля как по *Субконто*, так и по *КорСубконто* (Рисунок 486).

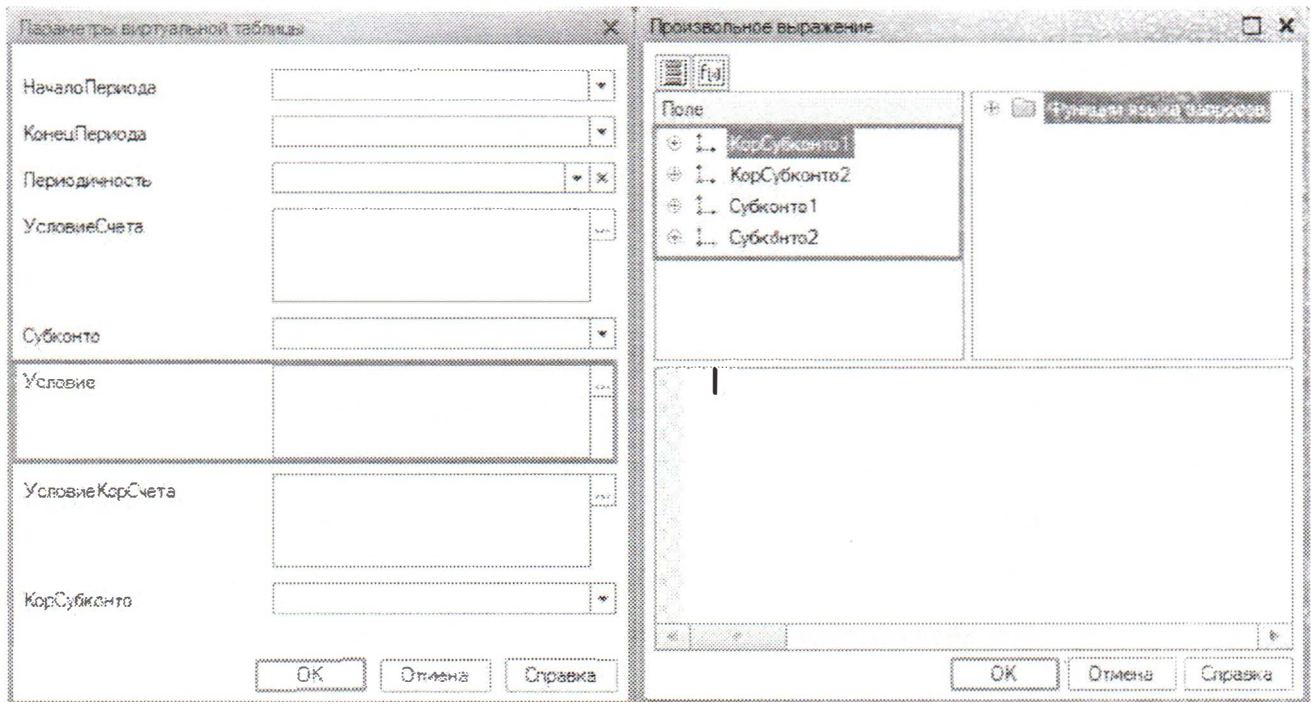


Рисунок 486

Однако для *КорСубконто* можно задать условие и отдельно. В таком случае результат будет выведен не по оборотам счета, а по оборотам корреспонденции счетов (между одним и другим счетом) (Рисунок 487).

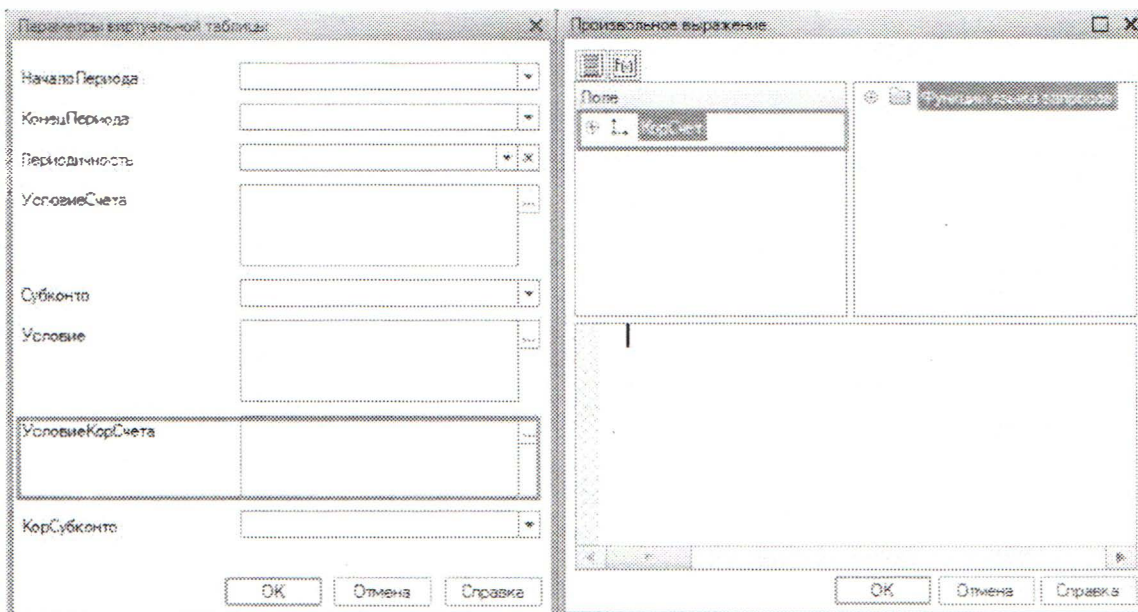


Рисунок 487

Что касается порядка работы с «КорСубконто», то он аналогичен принципам работы с параметром «Субконто»: точно по такому же принципу можно управлять аналитиками, их порядком и определять, какая из них будет КорСубконто1 и КорСубконто2. При этом обороты получаем как по дебету счета, так и по кредиту счета.

Рассмотрим параметры виртуальной таблицы «УправленческийОборотыДтКт» (Рисунок 488):

- НачалоПериода;
- КонецПериода;
- Периодичность;
- УсловиеСчетаДт;
- СубконтоДт;
- УсловиеСчетаКт;
- СубконтоКт;
- Условие.

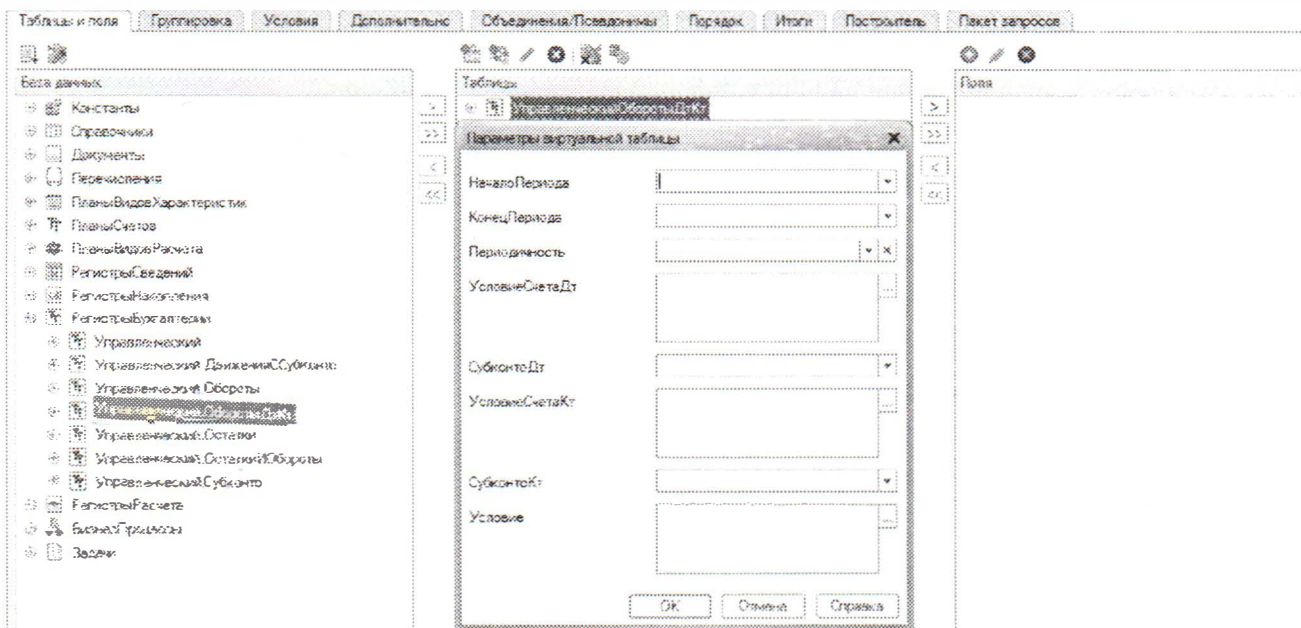


Рисунок 488

Из таблицы оборотов по дебету и кредиту мы можем получить обороты между определенным счетом дебета и определенным счетом кредита (если ранее мы получали обороты между управленческими счетами, независимо от того, где они находятся, то теперь мы можем указать параметры нахождения либо в дебете, либо в кредите).

Рассмотрим параметры виртуальной таблицы «Управленческий Остатки И Обороты» (Рисунок 489):

- НачалоПериода;
- КонецПериода;
- Периодичность;
- МетодДополнения (Движения; ДвиженияИГраницыПериода);
- УсловиеСчета;
- Субконто;
- Условие.

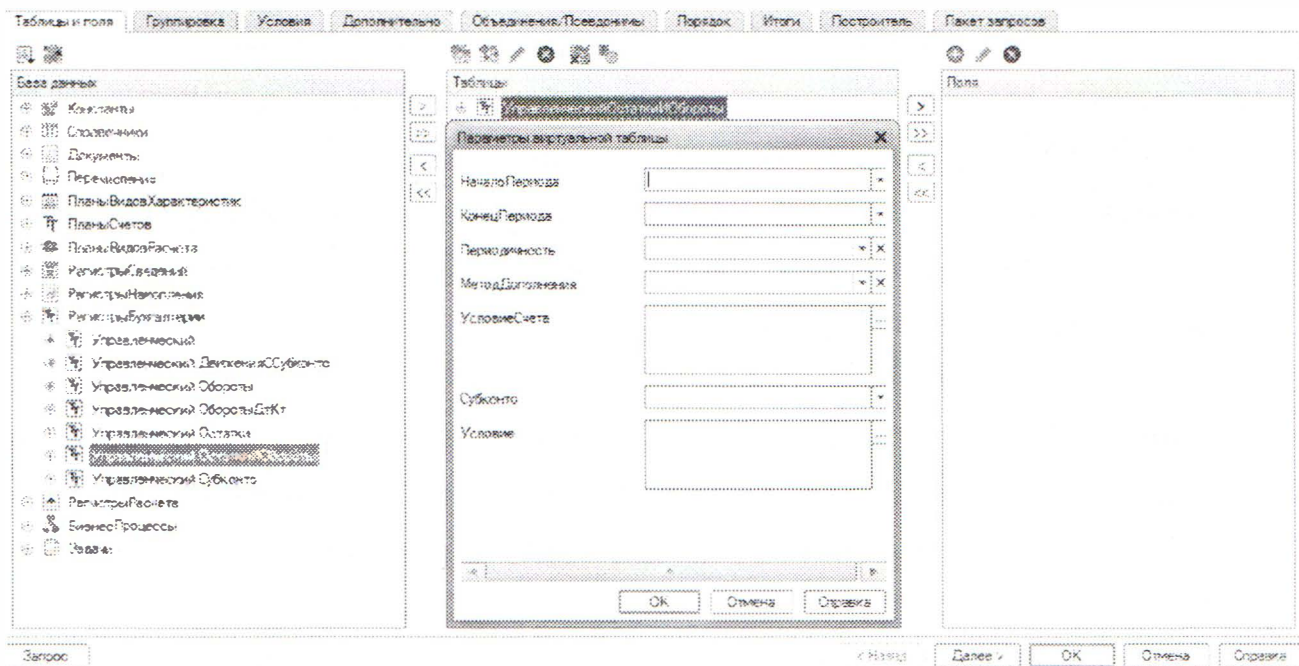


Рисунок 489

Параметр виртуальной таблицы «МетодДополнения» был рассмотрен ранее на примере регистров накопления и включает два вида дополнения: «Движения» и «ДвиженияИГраницыПериода». Соответственно, если во втором случае, мы укажем детализацию для регистратора, то будут записи с пустыми регистраторами для начального и конечного остатка, если укажем «Движения», то и их не будет.

Рассмотрим значения полей регистра «УправленческийДвиженияССубконто» (Рисунок 490):

- «<Имя измерения>» – поле, содержащее значения балансового измерения регистра. Количество таких полей равно количеству балансовых измерений, определенных для регистра как объекта конфигурации;
- «<Имя измерения>Дт» – поле, содержащее значения дебетового небалансового измерения регистра с именем, заданным в конфигурации;
- «<Имя измерения>Кт» – поле, содержащее значения кредитового небалансового измерения регистра с именем, заданным в конфигурации;
- «<Имя общего реквизита>» – поле, содержащее значения общего реквизита с именем, заданным в конфигурации. Такие поля создаются для общих реквизитов, являющихся разделителями (режим разделения данных – Разделять) с режимом использования разделяемых данных «НезависимоИСовместно», в которых и участвует данный регистр;
- «<Имя реквизита>» – поле, содержащее значения реквизита регистра с именем, заданным в конфигурации;
- «<Имя ресурса>» – поле, содержащее значения балансового ресурса регистра по именам ресурсов, как они заданы в конфигураторе;

- «<Имя ресурса>Дт» – поле, содержащее значения дебетового небалансового ресурса регистра по именам ресурсов, как они заданы в конфигураторе;
- «<Имя ресурса>ОборотКт» – поле, содержащее значения кредитового небалансового ресурса регистра по именам ресурсов, как они заданы в конфигураторе;
- «Активность» имеет тип «Булево». Содержит признак активности записи и влияния на получение итогов регистра;
- «ВидСубконтоДт<Номер субконто>» имеет тип «ПланВидовХарактеристикСсылка.<имя>». Количество таких полей зависит от максимального количества субконто на счете плана счетов. Номер вида субконто начинается с 1. Набор и порядок видов субконто соответствуют набору и порядку видов субконто у счета дебета;
- «ВидСубконтоКт<Номер субконто>» имеет тип «ПланВидовХарактеристикСсылка.<имя>». Количество таких полей зависит от максимального количества субконто на счете плана счетов. Номер вида субконто начинается с 1. Набор и порядок видов субконто соответствуют набору и порядку видов субконто у счета кредита;
- «Момент времени» – виртуальное поле, которое не хранится в информационной базе. Содержит объект «МоментВремени», который включает в себя дату и ссылку на документ-регистратор.
- «Период» имеет тип «Дата». Содержит дату записи. Совместно с полями «Регистратор» и «НомерСтроки» определяет положение данной записи на временной оси;
- «Регистратор» имеет тип «ДокументСсылка.<имя>». Содержит ссылку на документ, которому подчинена данная запись;
- «НомерСтроки» имеет тип «Число». Содержит уникальный номер данной записи в наборе записей регистра, подчиненных документу, указанному в поле «Регистратор»;
- «СубконтоДт<Номер субконто>» имеет тип «Характеристика.<имя>», тип этого поля определяется типом субконто. Содержит значение субконто дебета. Количество таких полей зависит от максимального количества субконто на счете плана счетов. Номер вида субконто начинается с 1. Набор и порядок видов субконто соответствуют набору и порядку видов субконто у счета дебета;
- «СубконтоКт<Номер субконто>» имеет тип «Характеристика.<имя>», тип этого поля определяется типом субконто. Содержит значение субконто кредита. Количество таких полей зависит от максимального количества субконто на счете плана счетов. Номер вида

субконто начинается с 1. Набор и порядок видов субконто соответствуют набору и порядку видов субконто у счета кредита;

– «СчетДт» имеет тип «ПланСчетовСсылка.<Имя>». Содержит ссылку на дебетуемый счет;

– «СчетКт» имеет тип «ПланСчетовСсылка.<Имя>». Содержит ссылку на кредитуемый счет.

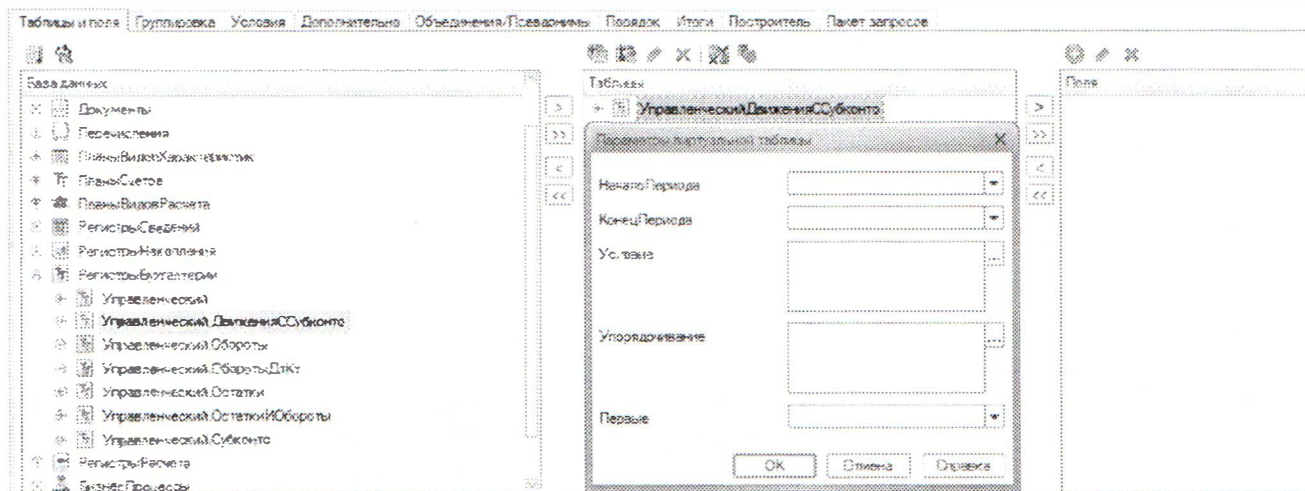


Рисунок 490

В этой таблице можно задать условия на счет дебета и счет кредита, а также на конкретные виды субконто в дебете и кредите.

5.3 Параметры виртуальных таблиц регистров расчета.

Регистры расчета представляют собой отдельный механизм в платформе «1С:Предприятие» и являются темой для отдельного рассмотрения. Это не является целью данной главы, поскольку регистры расчета имеют отличную от регистров накопления и бухгалтерии структуру виртуальных таблиц, разработанную под решение периодических начислений и перерасчетов. В связи с этим, останавливаться подробно на решении расчетных задач в этой главе мы не будем. Обратим внимание на параметры виртуальных таблиц регистров расчета.

Регистры расчета предназначены для организации учета результатов вычислений, выполненных расчетами, указанными в плане видов расчетов. Как правило, они используются для реализации начислений по оплате труда, оплате больничных листов, оплате дней отпуска.

Учетные записи регистра расчета могут влиять на состояние других записей регистра. Различают два характера взаимного влияния учетных записей регистра расчета: конкуренция за период действия и зависимость по базовому периоду. Подробно рассматривать их мы не будем (также отдельной темой является механизм перерасчетов).

Для работы с регистрами расчета предназначены «Регистры расчета» дерева конфигурации. Посмотрим, какие регистры расчета содержатся в каркасной конфигурации (Рисунок 491-492).

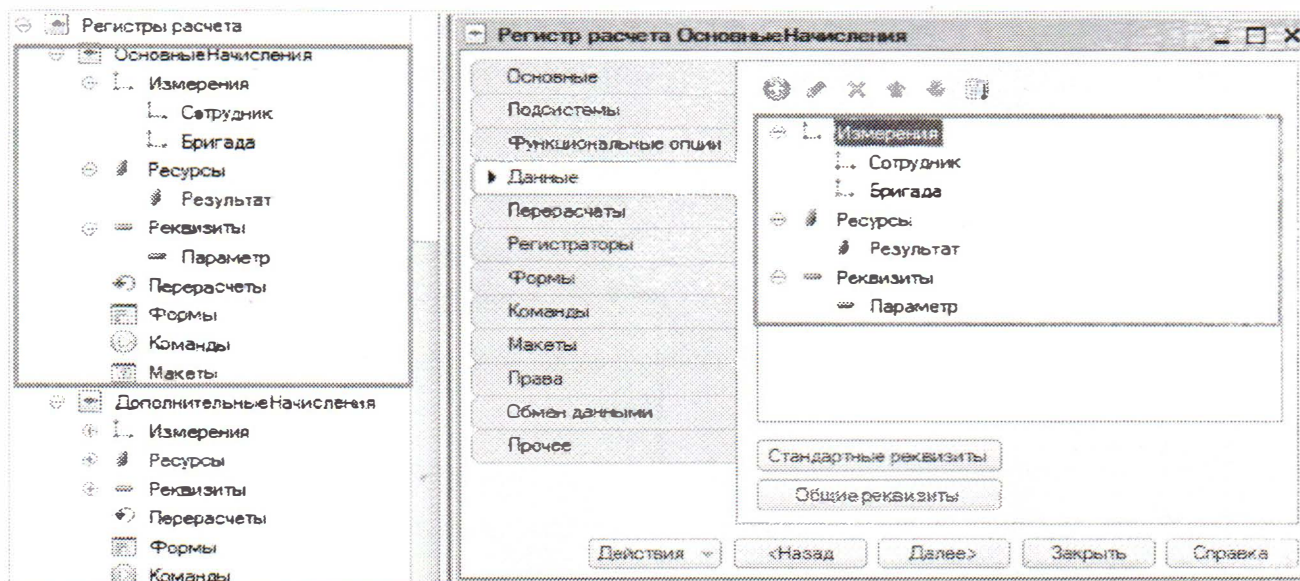


Рисунок 491

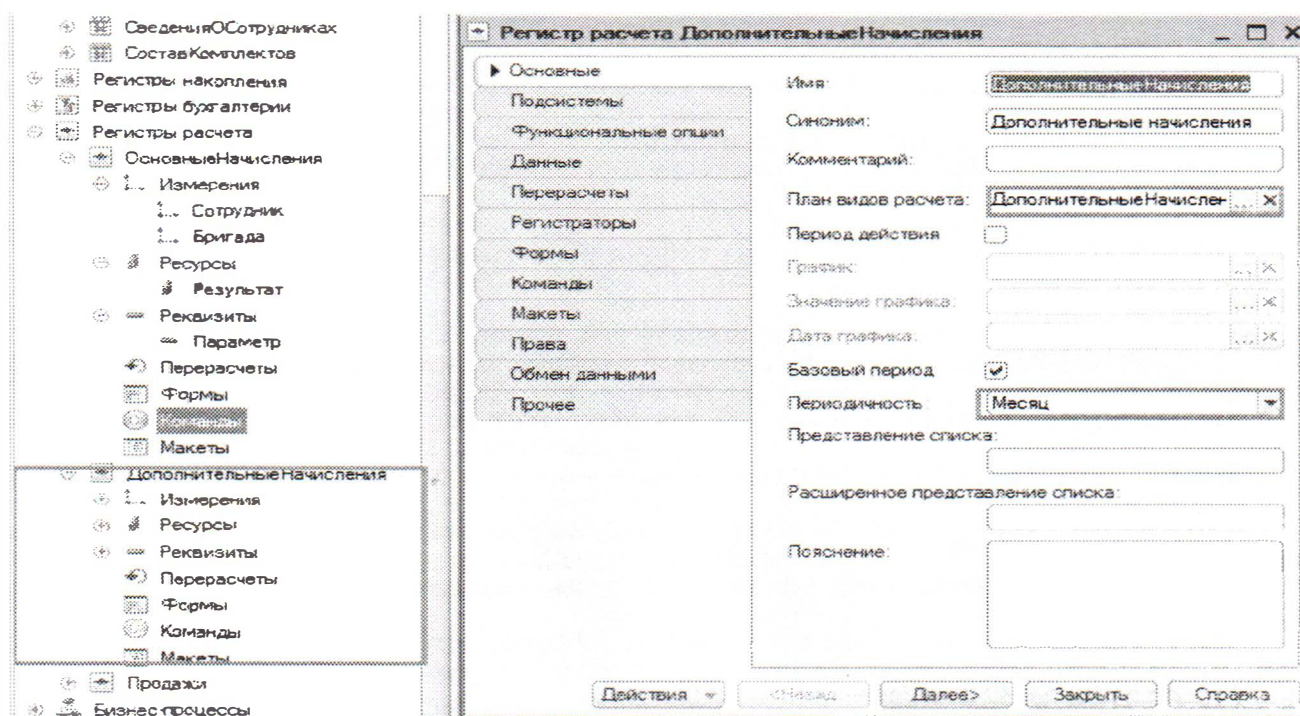


Рисунок 492

Как видно из свойств регистра «Дополнительные начисления», при редактировании регистра расчета определяется план видов расчета, поддержка периода действия и базового периода, периодичность.

Рассмотрим подробнее эти свойства (Таблица 8).

Таблица 8 – свойства регистра расчета

Свойство регистра расчета	Назначение
План видов расчета	Основная характеристика регистра – ссылка на соответствующий объект ПланВидовРасчета, который используется для создания видов расчетов, используемых в регистрах расчета. Ссылка на вид расчета – одно из основных свойств записей регистра расчета, которое позволяет придать качественные различия разным учетным записям регистра расчета.
Период действия	<p>Если свойство установлено, то взаимное влияние движений данного регистра принимает конкурирующий характер (например, начисление заработной платы и оплата больничного листа не могут выполняться одновременно). Такие расчеты взаимно исключают друг друга во времени, и система должна гарантировать, что ввод одного из них приведет к исключению другого.</p> <p>Регистр расчета с поддержкой периода действия имеет ограничение на количество записей с одинаковыми значениями измерений, периода регистрации и вида расчета.</p>
График	Свойство доступно, если установлено свойство Период действия. Свойство представляет ссылку на регистр сведений, в котором описывается временная схема исходных данных, участвующих в расчете. График следует указывать для тех расчетов, которые зависят от исходных данных, распределенных в пределах периода действия по определенному правилу. Например, это может быть график учета рабочего времени организации с разбивкой по дням, учет лекционных часов с разбивкой по часам и т. д.
Значение графика	Свойство доступно, если установлено свойство Период действия. В свойстве выбирается ресурс регистра сведений, определенного в свойстве График.
Дата графика	Свойство доступно, если установлено свойство Период действия. В свойстве выбирается измерение регистра сведений, определенного в свойстве График и имеющее тип Дата. По значению данного свойства осуществляется привязка к значениям ресурса регистра сведений, указанного в свойстве Значение графика.
Базовый период	Если свойство установлено, то устанавливается связанный характер взаимного влияния движений данного регистра. Примером связанных движений может служить связь расчета сумм начисления выплат по средней от сумм начисления в базовом периоде.
Периодичность	Определяет период, с которым регистрируются движения, и в пределах которого движения могут влиять друг на друга (для регистров, поддерживающих период действия).
Перерасчеты	<p>Подчиненные объекты регистра, которые позволяют задать правила взаимного влияния движений регистров. В группе Связь свойств объекта в свойстве Измерение регистра указывается основное измерение текущего регистра, которое следует пересчитать при изменении данных ведущих регистров, указанных в свойстве Данные ведущих регистров. Например, перерасчет суммы удержания по физическому лицу будет формироваться при изменениях начислений (оплата труда, премии).</p> <p>Если установлено свойство Базовый период, то формирование данных перерасчетов будет выполнено автоматически. Если свойство не установлено, то формирование данных перерасчетов должно производиться пользователем вручную (при проектировании следует разработать специальную форму ввода перерасчетов и механизм их выполнения).</p>

5.4 Контроль разрешения прав доступа на уровне записей. «Разрешенные».

Рассмотрим механизм контроля разрешения прав доступа на уровне записей. За это отвечает признак «Разрешенные» на вкладке «Дополнительно» конструктора запросов, благодаря которому в ходе исполнения запроса получаем данные, на которые есть права (Рисунок 493).

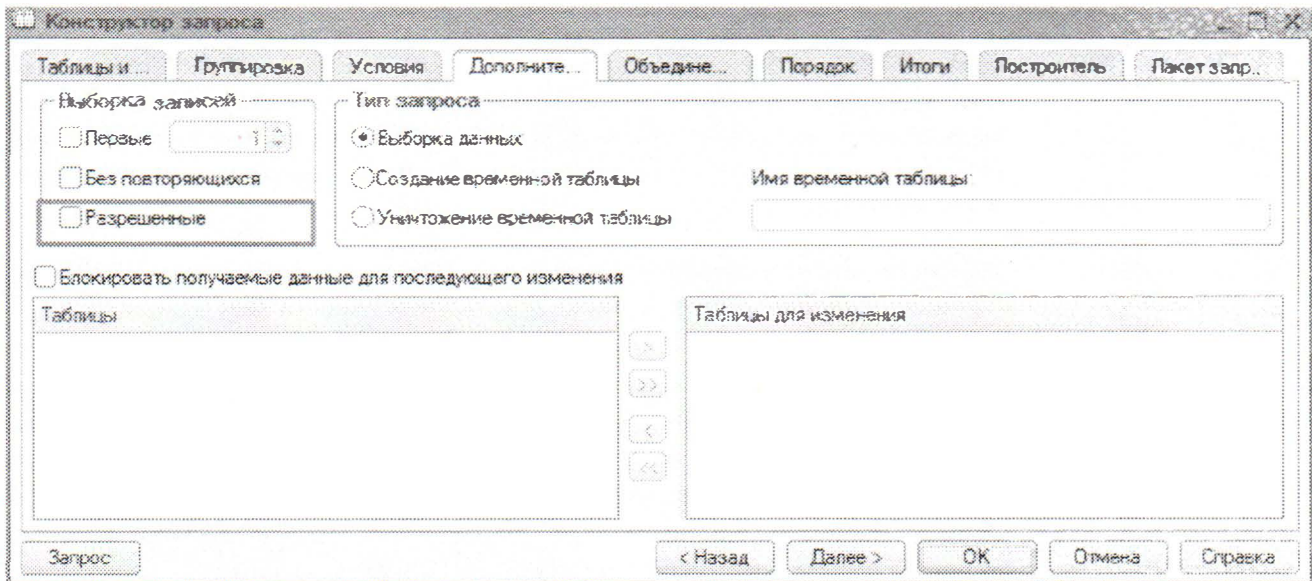


Рисунок 493

В случае если используется контроль разрешения прав на уровне записей и свойство «Разрешенные» находится в значении «Истина», мы получим только те записи результата запроса, на которые есть права. В случае если используется контроль разрешения прав на уровне записей и свойство «Разрешенные» находится в значении «Ложь», будет выведено сообщение об отсутствии прав работы с программой.

5.5 Блокировка данных для последующего изменения

Рассмотрим механизм блокировки данных для последующего изменения. За это отвечает отдельное свойство в конструкторе запросов – «Блокировать получаемые данные для последующего изменения» (Рисунок 494).

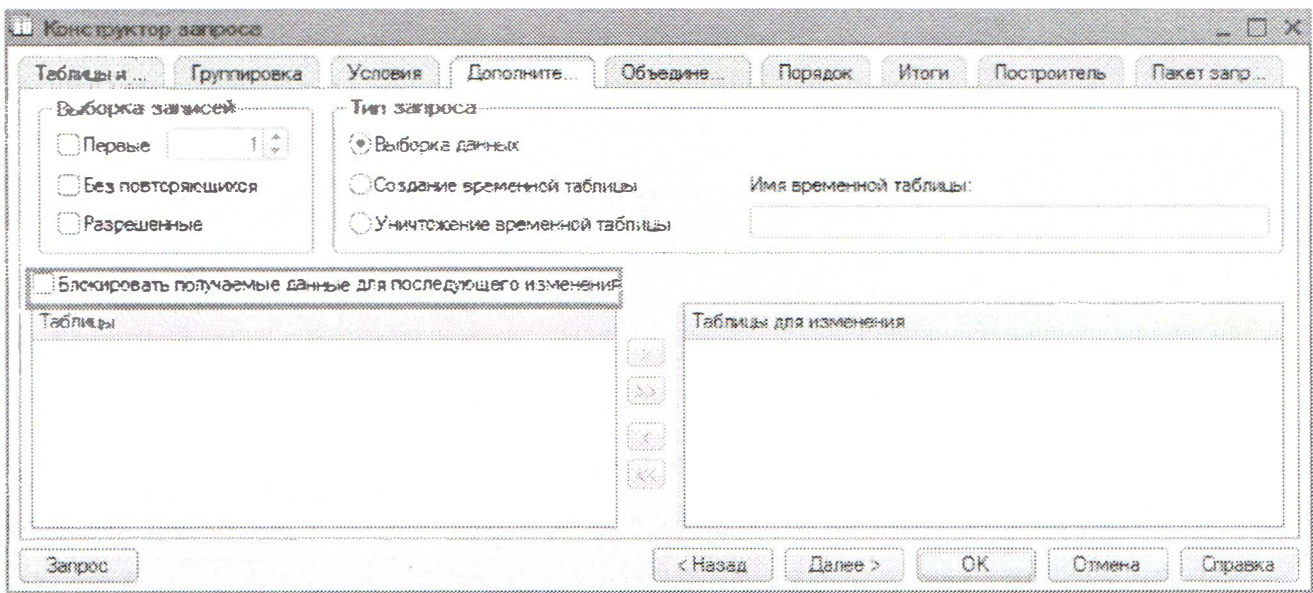


Рисунок 494

Рассмотрим работу механизма блокировок на примере таблицы остатков номенклатуры. Для этого откроем конструктор запроса и выберем поля таблицы «ОстаткиНоменклатуры» (Рисунок 495).

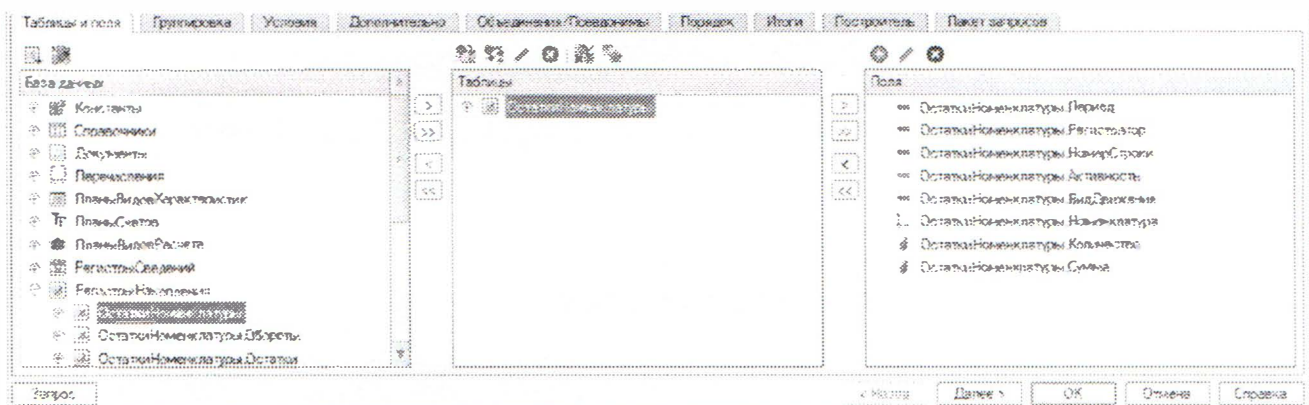


Рисунок 495

Далее перейдем на закладку «Дополнительно», где находится форма работы с блокировками, которая состоит из двух областей: общего списка таблиц (слева) и таблиц для изменения (справа). Перенесем таблицу «ОстаткиНоменклатуры» из области таблиц в область «Таблицы для изменения». Свойство «Блокировать получаемые данные для последующего изменения» должно находиться в значении «истина» (Рисунок 496).

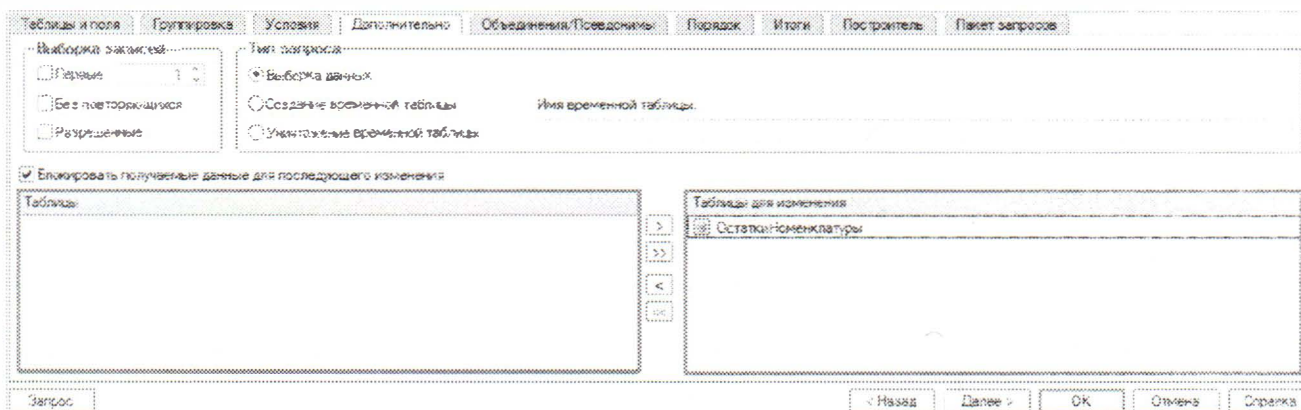


Рисунок 496

Механизм блокировки для последующего изменения актуален только для режима с автоматическими блокировками (в свойствах конфигурации нужно поставить «Режим управления блокировкой данных» в значение «Автоматический») (Рисунок 497).

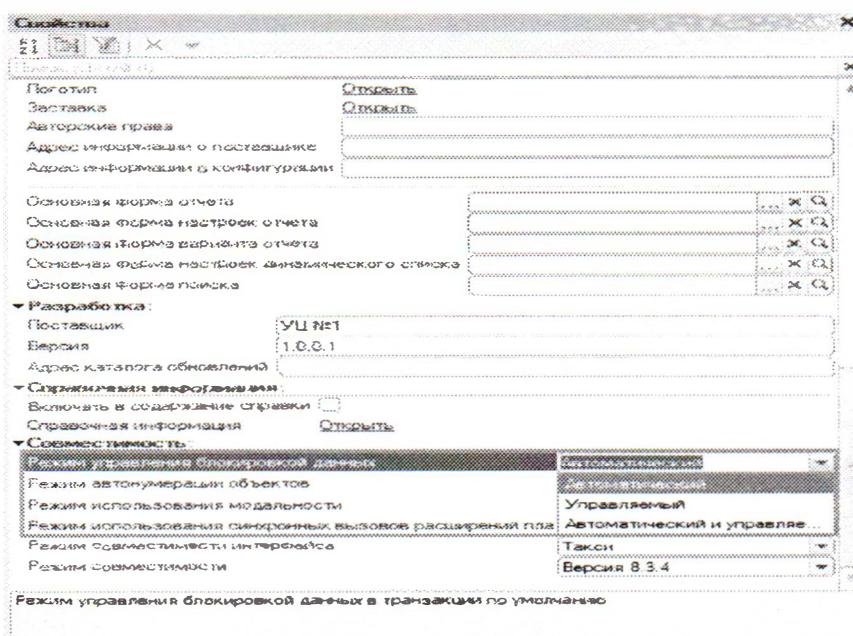


Рисунок 497

В случае если мы блокируем получаемые данные, таблица блокируется до конца транзакции. На примере проведения документа это работает следующим образом: в процессе выполнения обработки проведения запрашиваются остатки номенклатуры, далее до конца выполнения обработки таблица остатков блокируется для внесения изменения и никто из пользователей не может их редактировать. В случае с управляемыми блокировками пользователь сам управляет изменением записей таблиц.

Вы закончили изучение книги «Язык запросов 1С “ОТ ЗУБОВ”». Надеемся, что вам понравилось книга, и она поможет вам в вашей практической деятельности.

Предлагаем вам не останавливаться на достигнутом результате!

Записывайтесь на наши онлайн курсы и продолжайте движение вперед!

I. Базовый курс «Я любитель»

II. Основной курс «Я профессионал»

III . VIP курс «Я эксперт»

Ссылка для записи: http://work-1c.ru/Training_Programmer_1C/

Дополнительные курсы:

I. Язык запросов 1С (входит в VIP курс):

http://shop.work-1c.ru/Zapross/Zapross_prd_cbdsj

II. Компоновка данных (входит в VIP курс):

<http://work-1c.ru/1c-komponovka/>

III. Конвертация данных (входит в VIP курс):

<http://zayavka.work-1c.ru/1c-konvertacia/>

IV. Администрирование 1С (входит в VIP курс):

<http://administrirovanie1c.work-1c.ru/>

V. Разработка мобильных приложений (входит в VIP курс):

<http://mobileapps.work-1c.ru/>

VI. Зарплата и управление персоналом, редакция 3

http://shop.work-1c.ru/Video_HRM_Kurs/ZUP_3_1_Sales

VII. Расширение конфигурации «Базовый»/ «Эксперт»

<https://shop.work-1c.ru/Expansion/SalesExpansion>

VIII. Курс «Бухгалтерия для Программистов 1С»

https://shop.work-1c.ru/Video_Accaunt_Kurs/AccountingForProgrammers

IX. 1С:Бухгалтерия государственного учреждения 2.0

<https://shop.work-1c.ru/BGU/SalesBGU>

Звоните 8 800 2344322 – звонок бесплатный. Можете написать вопросы на email

mail@work-1c.ru

С уважением, проект «Верный старт в 1С».