# Занятие 8. Запросы. Табличная модель доступа к данным

### §8.1. Табличная модель доступа к данным. Запросы.

Наряду с рассмотренной выше объектной моделью доступа к данным, в системе «1С: Предприятие» реализована табличная модель. В основе ее – использование запросов к таблицам базы данных. В результате запроса возвращается набор данных, таблиц, которые можно анализировать и использовать. При помощи запроса программист никоим образом не может изменить данные. В этом отчасти преимущество запросов в системе «1С»: пользователь, программист не может случайно испортить информацию. В основе языка запросов лежит язык SQL с рядом ограничений (и, естественно, на русском языке).

Т.к. запрос производит чтение информации из таблиц базы данных, то, очевидно, что использовать запросы мы можем лишь на стороне сервера: либо в модуле формы при серверных вызовах, либо в модуле объекта (чаще всего). Текст запроса, его обработку и выполнение можно написать вручную, используя специальные конструкции, полный перечень которых приведен в Синтакс-помощнике. Однако в среде «1С: Предприятие» реализован более удобный инструмент для визуального создания запросов – конструктор запроса. Для вызова конструктора запроса в любой точке программного кода в модуле (формы, объекта) необходимо в контекстном меню выбрать либо «Конструктор запроса», либо «Конструктор запроса с обработкой результата». Конструктор запроса с обработкой результата отличается тем, что при его вызове автоматически формируется код специальных инструкций по выполнению запроса, передаче в него нужных параметров и обход результатов с целью их обработки и анализа; при вызове конструктора запроса без обработки результата производится формирование только текста запроса.

Решим следующую задачу. При поступлении товаров оказывается, что пользователь в табличную часть один и тот же товар может ввести несколько раз, при том даже по разным ценам. В результате при проведении документа в регистре накопления «**Остатки товаров**» будет сформировано несколько похожих записей. Необходимо эти записи сгруппировать, чтобы для одного товара было одно движение. А для регистра сведений

«**Цены поставщиков**» установить цену, максимальную для данного товара.

Отрываем процедуру обработки проведения в модуле объекта документа «**Приходная накладная**», и в начале процедуры запускаем «конструктор запроса с обработкой результата». На предложение создать новый запрос даем положительный ответ. В результате откроется следующее окно (рис. 42).

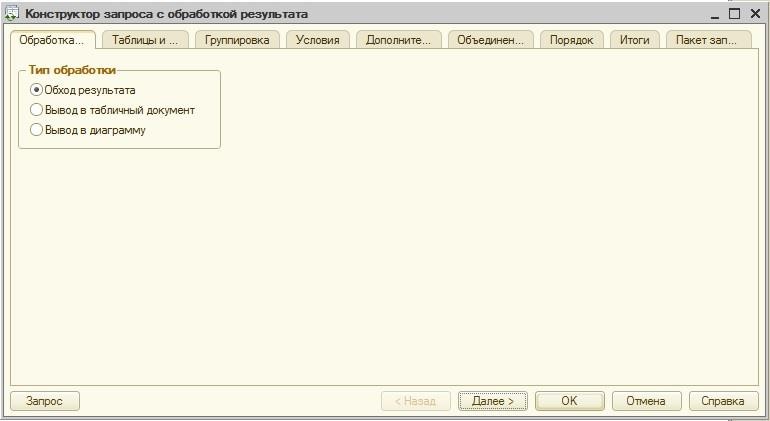


Рис. 42. Конструктор запроса с обработкой результата

Выбираем тип обработки «Обход результата» и переходим к следующей вкладке. На вкладке «Таблицы и поля» (рис. 43) слева представлена структура разрабатываемого приложения в виде таблиц базы данных и виртуальных таблиц. Это – источники данных. В нашей задаче источником данных будет таблица, представляющая табличную часть документа «**Приходная накладная**». Дело в том, что физически в базе данных реквизиты шапки (справочника, документа) хранятся в одной таблице, а под табличные части элементов создается дополнительная таблица, в которой указывается ссылка на «элемент- владелец» (справочника, документа), к которому данные записи относятся. Из выбранной таблицы необходимо выбрать поля: «**Номенклатура**», «**Цена**», «**Количество**», «**Сумма**» (рис. 43).

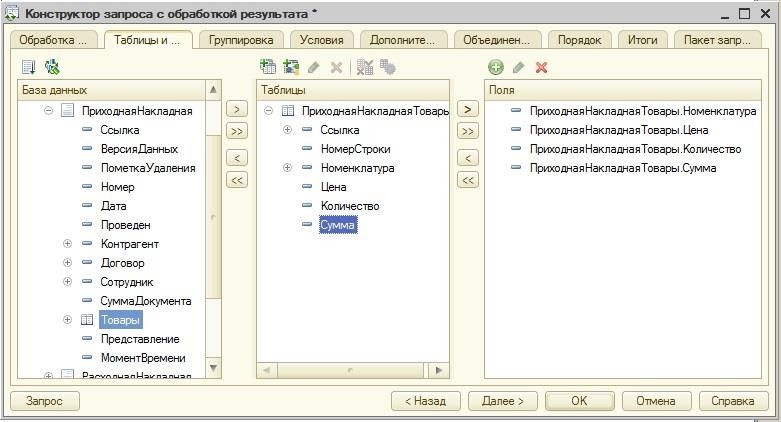


Рис. 43 Выбор источников данных и полей

На вкладке группировка (рис. 44) накладываются условия группировки: все поля разносятся по двум категориям: поля группировки и группируемые поля. Не должно возникать ситуации «висящих» полей, не принадлежащих ни той, ни другой группе.

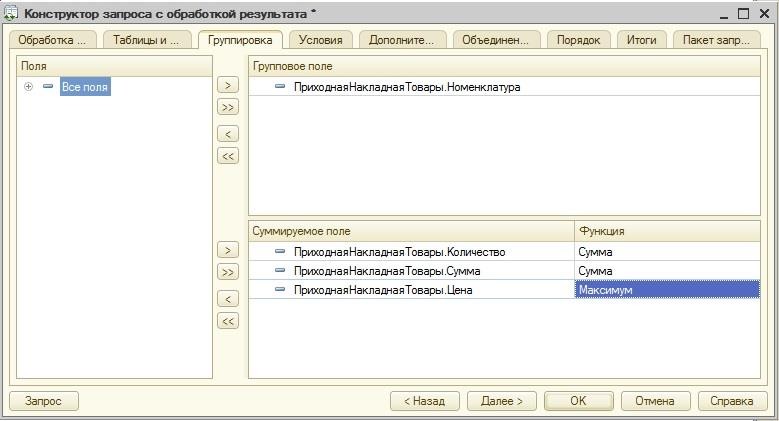


Рис. 44. Группировка полей в запросе

В нашей задаче полем группировки будет «**Номенклатура**» (одинаковые товары будут фигурировать в движениях один раз), все остальные поля – группируемые. К группируемым полям применяются агрегатные функции («Сумма», «Сумма различных»,

«Максимум» и т.д.). Для полей «**Количество**» и «**Сумма**» применяется агрегатная функция «Сумма» (для простого суммирования одинаковой номенклатуры, находящейся в различных строчках), а к полю «**Цена**» - функция «Максимум», для выбора наибольшей установленной цены для данной номенклатурной позиции.

На закладке «Условия» (рис. 45) можно наложить условия на поля запроса, тем самым ограничить число записей, возвращаемых в результате запроса. В нашей задаче мы хотим получить сгруппированные записи не по всем документам, а только по текущему. Для этого нужно наложить условие на реквизит «Ссылка», конкретную ссылку на документ будем передавать через параметр запроса (рис. 45).

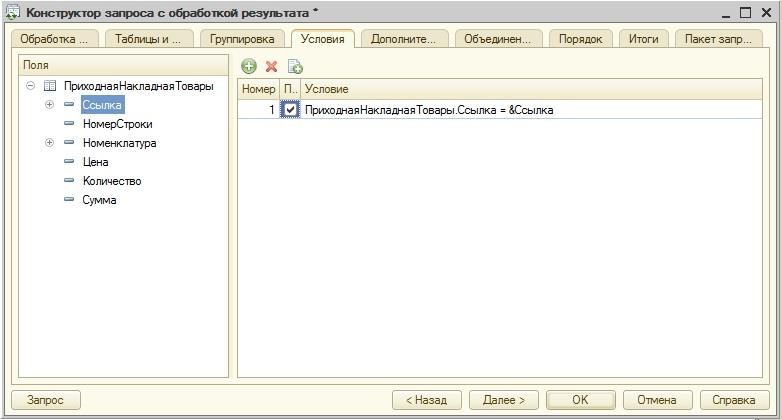


Рис. 45. Наложение условий в запросе

На вкладке «Объединения/Псевдонимы» можно задать псевдонимы для всех полей, возвращаемых в результате запроса. В нашем случае изменять ничего не нужно.

Для просмотра текста сформированного запроса можно нажать кнопку «Запрос». Для завершения формирования запроса нажимаем «ОК». В результате в модуле объекта сформировался следующий код:

|  |  |
| --- | --- |
| Запрос = Новый Запрос; Запрос.Текст =  "ВЫБРАТЬ  | ПриходнаяНакладнаяТовары.Номенклатура,  | МАКСИМУМ(ПриходнаяНакладнаяТовары.Цена) КАК Цена,  | СУММА(ПриходнаяНакладнаяТовары.Количество) КАК Количество,  | СУММА(ПриходнаяНакладнаяТовары.Сумма) КАК Сумма  |ИЗ  | Документ.ПриходнаяНакладная.Товары КАК ПриходнаяНакладнаяТовары  |ГДЕ  | ПриходнаяНакладнаяТовары.Ссылка = &Ссылка  |  |СГРУППИРОВАТЬ ПО  | ПриходнаяНакладнаяТовары.Номенклатура";  Запрос.УстановитьПараметр("Ссылка", Ссылка); РезультатЗапроса = Запрос.Выполнить(); ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать(); Пока ВыборкаДетальныеЗаписи.Следующий() Цикл  // Вставить обработку выборки ВыборкаДетальныеЗаписи КонецЦикла; | Л.14 |

В данном тексте вначале создается специальный объект типа «Запрос», после чего пишется сам текст запроса. Далее идет раздел задания параметров (УстановитьПараметр), в котором конструктор автоматически определяет параметры, передаваемые в запрос:

слева указывается имя параметра в строковом представлении, справа – его значение. Если значение параметра, сформированное автоматически, оказалось неправильным, то его нужно исправить. Далее идет инструкция, выполняющая запрос, после чего следует передача результата запроса: в таблицу, диаграмму, или выборку. Выборка используется для обхода каждой записи в результате запроса. Используя метод ***Выбрать()*** система позиционируется на начале выборки. Для перехода к каждому следующему элементу выборки и считыванию значений конкретных полей используется метод ***Следующий()***. Соответственно, представленный цикл осуществляет переход к каждому следующему элементу выборки до конца. Внутри данного цикла мы можем обращаться к полям запроса как к полям выборки, например, **ВыборкаДетальныеЗаписи.Номенклатура**.

В нашей задаче параметр, передаваемый в запрос верный: мы передаем ссылку на текущий обрабатываемый документ. Цикл обхода элементов выборки будет выглядеть следующим образом (жирным выделены изменения в формировании движений, по сравнению с предыдущим вариантом):

|  |  |
| --- | --- |
| Пока ВыборкаДетальныеЗаписи.Следующий() Цикл Движение = Движения.ОстаткиТоваров.Добавить();  Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.Период = Дата;  Движение.Номенклатура = **ВыборкаДетальныеЗаписи**.Номенклатура; Движение.Количество = **ВыборкаДетальныеЗаписи**.Количество; Движение.Стоимость = **ВыборкаДетальныеЗаписи**.Сумма;  Движение = Движения.ЦеныПоставщиков.Добавить(); Движение.Период = Дата;  Движение.Контрагент = Контрагент;  Движение.Номенклатура = **ВыборкаДетальныеЗаписи**.Номенклатура; Движение.Цена = **ВыборкаДетальныеЗаписи**.Цена;  КонецЦикла; | Л.15 |

После цикла, не забываем оставить возможность записи движений в регистры:

|  |  |
| --- | --- |
| Движения.ОстаткиТоваров.Записывать = Истина; Движения.ЦеныПоставщиков.Записывать = Истина; | Л.16 |

Весь оставшийся предыдущий код процедуры обработки проведения, удаляем. Запускаем систему и проверяем данный механизм на практике. Обратите внимание,

что запрос используется лишь для получения, группировки и обработки данных. При этом запрос работает намного быстрее, чем обращение через точку (объектная модель). Но для записи данных все равно используется объектная модель.

### Самостоятельная работа №3:

Аналогичным образом выполните группировку строк в документе «**Расходная накладная**» перед записью в регистры накопления. Измените процедуру проведения для документа «**Расходная накладная**». Учтите тот факт, что по табличной части «**Товары**» формируются движения в регистрах «**Остатки товаров**» и «**Продажи**», а по табличной части «**Услуги**» – только по регистру «**Продажи**».

### 8.2. Контроль остатков на складе

При проведении документа «**Расходная накладная**» необходимо проверять, хватает ли товара, который мы хотим продать или нет. В случае если товара хватает, документ проводится, иначе – выдается сообщение о том, какого товара и в каком количестве недостаточно и документ не проводится. Для решения поставленной задачи необходимо сформировать запрос, который будет возвращать количество продаваемого товара и остатки по данным номенклатурным позициям.

Исправим имеющийся запрос в модуле объекта документа «**Расходная накладная**». Для этого спозиционируемся на уже имеющемся тексте первого запроса (по «**Товарам**») и вызовем «Конструктор запроса».

Группировку строк табличной части оставляем т.к. если фигурирует один и тот же товар много раз, необходимо знать, сколько ВСЕГО указанного товара необходимо продать, а затем проверить есть ли он на складе.

Воспользуемся механизмом пакетных запросов, для контроля остатков по данным сгруппированным записям. Для этого перейдем на вкладку «Дополнительно» (см. рис. 42) выберем пункт «Создание временной таблицы», укажем «**ТоварыТЧ**» в качестве имени временной таблицы. Таким образом, после выполнения первого запроса из пакета (группировки) будет создана еще одна таблица – источник для следующих запросов в пакете.

Переходим на вкладку «Пакет запросов» (см. рис. 42) и добавляем еще один запрос.

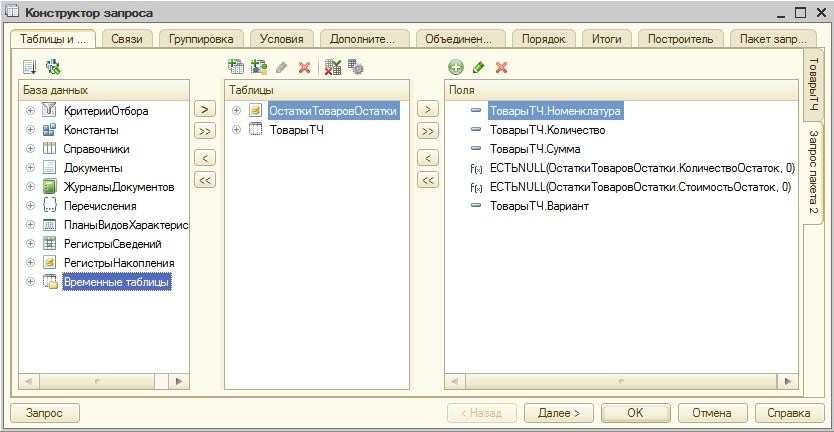


Рис. 46 Конструктор запроса в режиме пакетных запросов

Для перехода между двумя запросами из пакета справа имеются соответствующие вкладки (рис. 46). Во втором запросе к источникам данных добавился еще один источник

* «Временные таблицы». Выбираем временную таблицу «**ТоварыТЧ**» в качестве одного источника. В качестве второго источника выбираем виртуальную таблицу

«**ОстаткиТоваровОстатки**» регистра накопления «**Остатки товаров**». Данная виртуальная таблица содержит сгруппированные итоговые записи по остаткам товаров на дату, которая будет передаваться ей в качестве параметра.

В качестве полей, возвращаемых запросом из таблицы «**ТоварыТЧ**» выбираем все поля, а из виртуальной таблицы остатков – «**КоличествоОстаток**» и

«**СтоимостьОстаток**».

Для виртуальной таблицы необходимо настроить параметры, для того, чтобы ограничиться остатками только тех товаров, которые мы продаем на дату документа

«Расходная накладная». В противном случае (если не задать параметры) будут:

* получены остатки на текущую дату (это плохо при создании документов

«задним числом» или при перепроведении старых документов);

* получены остатки по всем товарам, т.е. все записи регистра. Это существенно снижает быстродействие: представьте, что вы продаете 5 номенклатурных позиций, а запрос получит остатки по тысячам позиций, т.е. вернет излишнюю информацию.

Для задания параметров используется кнопка  (рис. 47).

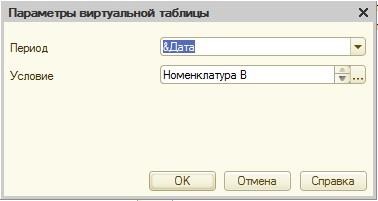


Рис. 47. Параметры виртуальной таблицы Остатки

В поле «Период» укажем: ***&Дата***. Тем самым из модуля объекта в запрос будет передана дата документа, на которую необходимо проверить остатки. В поле «Условия» мы можем указать дополнительные условия на измерения регистра накопления. Это необходимо для того, чтобы ограничить число возвращаемых результатов. В нашей задаче мы должны ограничиться получением остатков лишь по той номенклатуре, которую в данный момент продаем, а не по всей, что есть. Запишем следующее условие, нажав :

|  |  |
| --- | --- |
| Номенклатура В (ВЫБРАТЬ  ТоварыТЧ.Номенклатура  ИЗ  ТоварыТЧ КАК ТоварыТЧ) | Л.17 |

Данным «внутренним» запросом мы выбираем сгруппированную ранее номенклатуру из временной таблицы.

К полям «**КоличествоОстаток**» и «**СтоимостьОстаток**» дополнительно применяем функцию ***ЕстьNULL()***. Это необходимо для того, чтобы при отсутствии значений (нет записей об остатках) запрос возвращал не NULL, а некоторое число (в нашем случае ноль). В противном случае, мы не сможем сравнить требуемое для продажи количество товара с NULL. Для применения функций языка запросов, а также любого иного редактирования полей, возвращаемых запросом, используется . Задаем следующие функции вместо полей «**КоличествоОстаток**» и «**СтоимостьОстаток**» соответственно:

|  |  |
| --- | --- |
| ЕстьNULL(ОстаткиТоваровОстатки.КоличествоОстаток, 0)  ЕстьNULL(ОстаткиТоваровОстатки.СтоимостьОстаток, 0) | Л.18 |

В запросе присутствуют более одного источника (таблицы), поэтому их необходимо связать для соответствия записей (полное, левое и т.д. соединения). В нашем случае

необходимо использовать левое соединение таблиц: выбираются все записи из

«**ТоварыТЧ**» (продаваемые товары), и для тех записей, для которых есть соответствие в

«**ОстаткиТоваровОстатки**» (т.е. есть остатки по продаваемым товарам) происходит присоединение этих записей из второй таблицы (остатков). Связь производится по полю

«**Номенклатура**» (рис. 48).

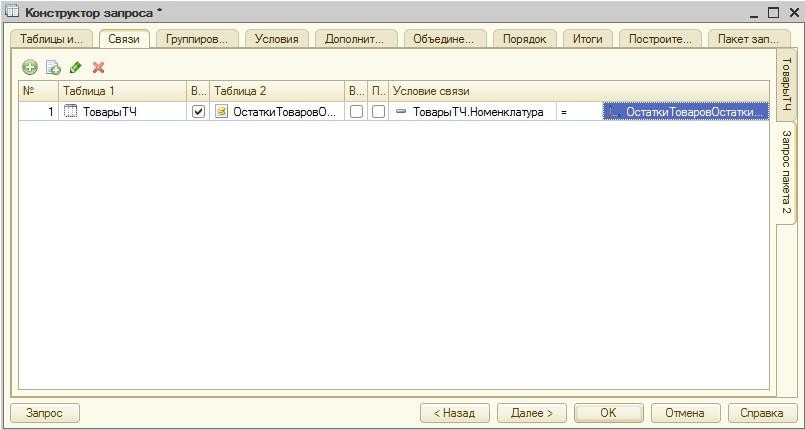


Рис. 48. Настройка левого соединения таблиц-источников

В заключении на вкладке «Объединения/Псевдонимы» отредактируем имена полей для остатков количества и стоимости: «**КоличествоОстаток**» и «**СтоимостьОстаток**». На этом формирование запроса завершено, возвращаемся в модуль объекта.

Т.к. производилось редактирование лишь текста запроса, то новые параметры необходимо указать самостоятельно. Так, после установки параметры «Ссылка», добавим следующий текст:

|  |  |
| --- | --- |
| Запрос.УстановитьПараметр("Дата", Дата); | Л.19 |

Затем исправим цикл обхода результата, в котором для каждого элемента выборки результата запроса необходимо произвести контроль остатков. Для этого перед формированием движений добавим следующую проверку:

|  |  |
| --- | --- |
| Если ВыборкаДетальныеЗаписи.КоличествоОстаток < ВыборкаДетальныеЗаписи.Количество Тогда  Сообщить("Не хватает "+ВыборкаДетальныеЗаписи.Номенклатура+" в количестве "+  (ВыборкаДетальныеЗаписи.Количество- ВыборкаДетальныеЗаписи.КоличествоОстаток));  Отказ = Истина; Движения.Продажи.Записывать = Ложь; Движения.ОстаткиТоваров.Записывать = Ложь;  Продолжить; КонецЕсли; | Л.20 |

В листинге Л.20 проверяется, что в остатках товара меньше, чем требуется. Если это так, то формируется сообщение, в котором указано, сколько и какого товара не хватает. Далее идет инструкции прерывающие транзакцию проведения документа и запрещающие запись в регистры. В конце команда ***Продолжить*** прерывает текущую итерацию цикла и переходит к следующему элементу выборки (чтобы проверить остатки по всем продаваемым товарам).

Отметим, что при проведении продажи в регистр накопления «**Остатки товаров**» записывается сумма продажи. Это не совсем верно: ведь мы должны списывать себестоимость (именно она хранится в регистре), а не сумму продажи. Себестоимость в самом простейшем случае рассчитывается по методу средневзвешенной скользящей:

|  |  |
| --- | --- |
| СебестоимостьПродажи =  (КоличествоПродажи/КоличествоОстаток)\*СебестоимостьОстаток | Л.21 |

Поэтому исправим движение по себестоимости (стоимости) в регистре «**Остатки товаров**» (в «**Продажах**» все верно, мы указываем выручку) следующим образом:

|  |  |
| --- | --- |
| Движение.Стоимость = ВыборкаДетальныеЗаписи.Количество /  ВыборкаДетальныеЗаписи.КоличествоОстаток \* ВыборкаДетальныеЗаписи.СтоимостьОстаток; | Л.22 |

Запустите систему и проверьте механизм контроля остатков. Следует отметить, что при работе с регистрами в большинстве случаев удобнее и эффективнее работать именно с виртуальными, а не реальными таблицами. Т.к. система формирует их «на лету», то можно достаточно гибко управлять их содержимым через настройку параметров. Использование параметров виртуальных таблиц обязательно: чем больше параметров мы укажем, тем сильнее ограничим возвращаемые данные лишь необходимыми, тем быстрее система обработает запрос. Согласно концепции системы, вначале формируются виртуальные таблицы, а только потом запрос обрабатывает данные. Поэтому чем меньше исходных данных, тем эффективнее работа самого запроса.

### Темы для самостоятельного изучения:

1. Виртуальные таблицы. Параметры виртуальных таблиц.
2. Вложенные запросы.
3. Менеджер временных таблиц. Уничтожение временных таблиц.
4. Упорядочение результатов запроса.
5. Формирование итогов в запросе. Обход результата запроса по полям группировок.
6. Вычисляемые поля в запросе.