

Контроль остатков при проведении документа «Расходная накладная»

При проведении документов, оформляющих расход товаров, важно не допустить возможность списания отсутствующего количества товара. Проблемой этой задачи является то, что при неправильной организации обработки данных можно не только существенно замедлить быстродействие приложения, но и создать немалую нагрузку на сетевые ресурсы, особенно в многопользовательском режиме. А запросы к файлам базы данных, расположенным на жестком диске, в цикле в многопользовательском режиме вообще может блокировать работу на многих рабочих местах.

Как правило, в табличной части документа фигурирует список товаров, а не какая-то единичная позиция, и обработка списания товаров осуществляется построчно в цикле по табличной части документа. При упущении из виду факта, что жесткий диск — самое медленное устройство компьютера, а значит количество и частота обращений к данным на этом устройстве должно быть минимальным, можно получить вышеописанную проблему. В следующем примере демонстрируется одна из реализаций такой задачи: списание товаров осуществляется путем проведения документа «Расходная накладная», при этом документ делает вид движения «Расход» по регистру «Товары на складах».

N	TMLC	Цена	Количество	Стоимость
1	Стол письменный	500	3	1 500

Рис. 1. Экранная форма документа «Расходная накладная»

Проведение: Разрешить
Оперативное проведение: Разрешить
Удаление движений: Удалять автоматически
Регистры складов
Регистры накопления
Товары на складах

Регистры: РегистрНакопления.ТоварыНаСкладах
Реквизиты документа: Дата
Тип движения регистра: Приход, Расход
Табличная часть: Номенклатура
Поле: TMLC, Склад, Количество
Выражение: ТекСтрокаНоменклатура.TMLC, Склад, ТекСтрокаНоменклатура.Количество

Рис.2. Организация движений документа по регистрам через Конструктор движений

Как правило, задача решается следующим образом: при проведении документа сравниваются хранящиеся в базе данных остатки по данной номенклатуре с количеством, запрашиваемом в документе. При этом товарные остатки удобно получить, обратившись к специальной виртуальной таблице **Остатки** соответствующего регистра накопления (в нашем случае «Товары на складах»), т. к. через механизм виртуальных таблиц остатков платформа быстро просчитывает итоги-остатки по регистру накопления с использованием нужных для разработчика параметров. В случае контроля списания остатков товаров остатки нужно получать на **дату списания** по указанному в документе **складу** и **материально-ответственному лицу** для **перечня номенклатуры**, указанного табличной части документа.

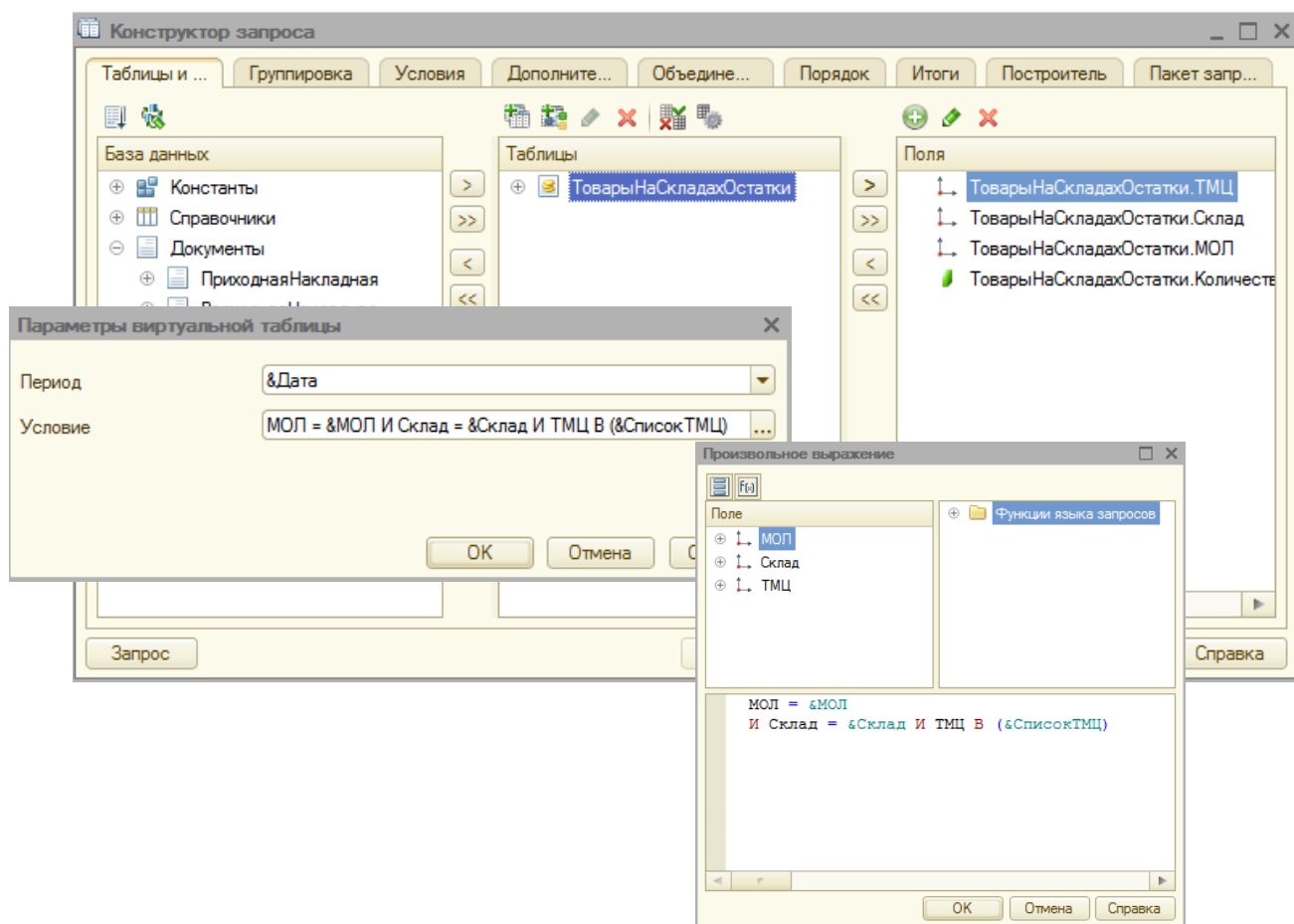


Рис.3. Задание параметров виртуальной таблицы «ТоварыНаСкладахОстатки» через механизм платформы «Конструктор запросов»

Параметрами виртуальной таблицы являются: &Дата — для указания значения даты, на которую должны быть сформированы остатки в виртуальной таблице; &МОЛ — в параметр должна быть передана ссылка на соответствующий элемент справочника Сотрудники; &Склад - в параметр должна быть передана ссылка на соответствующий

элемент справочника Склады; &СписокТМЦ - в параметр должен быть передан список номенклатуры, указанный в табличной части документа (иными словами, в параметр передается множество значений).

Запись данных документа по регистрам осуществляется в процедуре ОбработкаПроведения(), находящейся в так называемом **модуле объекта**, обратиться к которому можно через контекстное меню объекта в дереве метаданных. Модуль объекта предназначен для обработки стандартных событий, связанных с данным объектом.

Все изменение данных в регистрах осуществляются через специальный объект, называемый **набор записей**. У каждого документа существует свойство **Движения**. В этом свойстве содержатся **наборы записей регистров**, по которым документ делает движения.

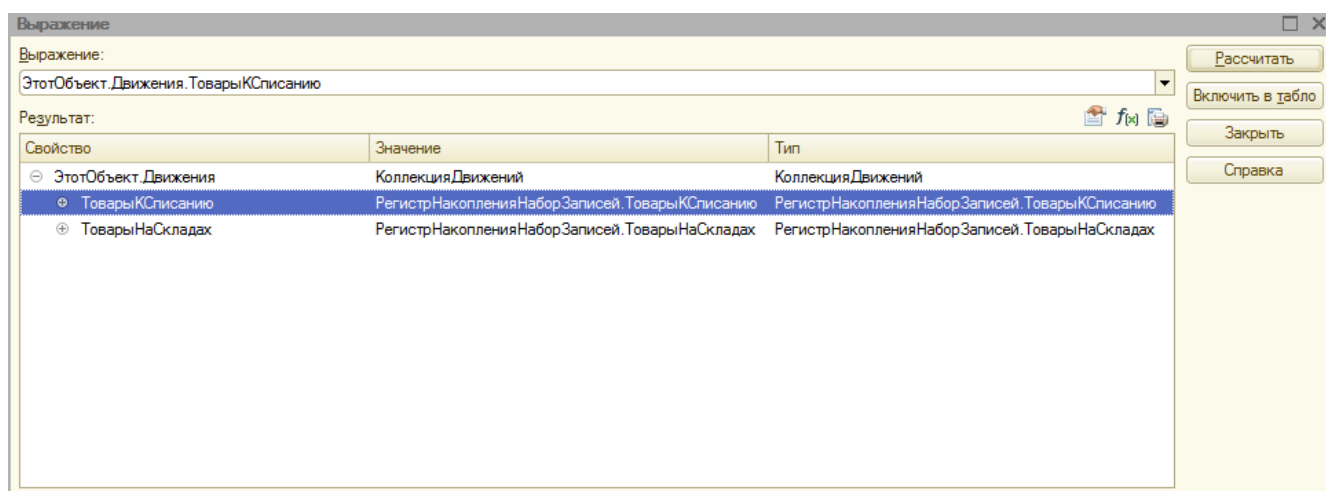


Рис.4. Движения документа « Расходная накладная »

После записи документа, платформа осуществляет автоматическую фиксацию всех данных из наборов записей, хранящихся в свойстве **Движения в регистры**.

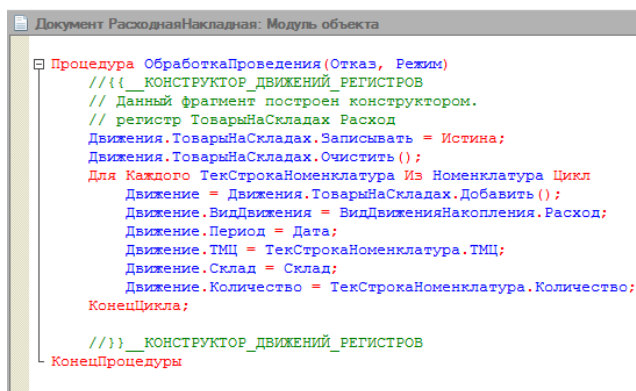


Рис.5 Процедура проведения документа в модуле объекта

Как видно из рисунка выше, информация о списываемой номенклатуре заносится в наборы данных в цикле, а это означает, что, если осуществлять контроль остатков в каждой итерации, обращаясь с запросом по текущей номенклатуре к жесткому диску, то можно и получить выше описанную ситуацию нерациональной организации приложения.

Решение проблемы заключается в одноразовой «выгрузке» через запрос к базе данных всей необходимой информации обо всей номенклатуре табличной части с жесткого диска в такие временные объекты, которые будут находиться в оперативной памяти и дальнейшей работе с ними. Для этого в стандартной схеме обработке запросов чаще всего используются такие объекты, как **ТаблицаЗначений** и **ВыборкаДанных**.



Рис.6. Стандартная схема обработки запроса

ТаблицаЗначений — это временный объект, предназначенный для хранения данных в табличном виде. Этот объект с помощью своих методов позволяет манипулировать своими строками и предоставляет доступ к коллекции колонок. .

ВыборкаДанных — это временный объект, позволяющий обойти данные, попавшие в результирующий набор, полученный по запросу к базе данных, и обработать их в цикле. Как видно из рисунка выше, все выше перечисленные временные объекты дают возможность обхода своих данных в цикле. Это и используется для обработки контроля списания остатков для каждой номенклатурной позиции документа.

Ниже на рис.7 представлен возможный код процедуры ОбработкаПроведения().

```

Процедура ОбработкаПроведения(Отказ, Режим)
Запрос = Новый Запрос;      // Создание объекта Запрос
Запрос.Текст =              // Определение текста запроса
"ВЫБРАТЬ
|   ТоварыНаСкладахОстатки.ТМЦ КАК ТМЦ,
|   ТоварыНаСкладахОстатки.Склад КАК Склад,
|   ТоварыНаСкладахОстатки.МОЛ КАК МОЛ,
|   ТоварыНаСкладахОстатки.КоличествоОстаток КАК КоличествоОстаток
| ИЗ
|   РегистрНакопления.ТоварыНаСкладах.Остатки (
|       &Дата,
|       МОЛ = &МОЛ
|       И Склад = &Склад
|       И ТМЦ В (&СписокТМЦ)) КАК ТоварыНаСкладахОстатки";

Запрос.УстановитьПараметр("Дата", Дата);          // Присвоение значений параметрам запроса   Дата, МОЛ, Склад
Запрос.УстановитьПараметр("МОЛ", МОЛ);
Запрос.УстановитьПараметр("Склад", Склад);

СписокТМЦ = Новый СписокЗначений;                // Создание временного объекта СписокЗначений
СписокТМЦ.ЗагрузитьЗначения(Номенклатура.ВыгрузитьКолонку("ТМЦ")); // Выгрузка колонки табличной части в список значений
Запрос.УстановитьПараметр("СписокТМЦ", СписокТМЦ ); // Установка списка значений в качестве параметра

тзРезультатаЗапроса = Запрос.Выполнить().Выгрузить(); // Выгрузка результата запроса в таблицу значений
Движения.ТоварыНаСкладах.Записывать = Истина;      // Движения записывать в регистр
Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
    НайденнаяСтрока = тзРезультатаЗапроса.Найти(ТекСтрокаНоменклатура.ТМЦ, "ТМЦ");
    НаличиеТовара = НайденнаяСтрока.КоличествоОстаток - ТекСтрокаНоменклатура.Количество;
    Если НаличиеТовара > 0 Тогда
        Движение = Движения.ТоварыНаСкладах.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.ТМЦ = ТекСтрокаНоменклатура.ТМЦ;
        Движение.Склад = Склад;
        Движение.Количество = ТекСтрокаНоменклатура.Количество;
    Иначе
        Сообщить("Не достаточно товара "+ ТекСтрокаНоменклатура.ТМЦ.Наименование+ " в количестве "+
        НаличиеТовара+" "+ ТекСтрокаНоменклатура.ТМЦ.ЕдХранения);
        Отказ = Истина;
    КонецЕсли
КонецЦикла;
КонецПроцедуры

```