

## Практическая работа № 13

### *Оптимизация проведения документа «Оказание услуги» (3:20)*

После изучения предыдущего занятия вы уже достаточно знакомы с языком запросов и можете приступить к одному из самых важных занятий – к оптимизации документа **ОказаниеУслуги**, в частности – к полному изменению обработчика события **ОбработкаПроведения**.

Зачем это нужно? Этому есть три причины.

Во-первых, в обращении к событию **ОбработкаПроведения** используется обращение через точку, что может сильно замедлить скорость проведения при больших объемах табличной части документа.

Во-вторых, руководство нашей фирмы решило прекратить ручной ввод стоимости расходуемых материалов и перейти на автоматический расчет «по среднему».

В-третьих, при проведении документа необходимо контролировать остатки расходуемых товаров на складе. Если товаров не хватает – выдавать предупреждение и не проводить документ.

Поэтому в нашей работе три цели:

1. Повышение скорости выполнения процедуры;
2. Автоматическое определение стоимости расходуемых материалов при проведении документа;
3. Разделение алгоритма проведения документа на оперативный и неоперативный режимы и контроль остатков в оперативном проведении документа.

Если алгоритм проведения документа использует только те данные, которые присутствуют в реквизитах документа (и его табличных частях), вполне достаточно использовать конструктор движений документа.

Если же в алгоритме проведения необходимо анализировать дополнительные реквизиты объектов, ссылки на которые содержатся в документе, а также использовать результаты расчета итогов регистров, следует использовать запросы для более быстрой выборки.

Механизм запросов лучше «читает» информационную базу и может за один раз выбрать только нужные данные. Поэтому в типовых решениях

вы не увидите использование объекта встроенного языка **СправочникВыборка. <имя>**. Вместо этого используются запросы к БД.

## Повышение скорости проведения

Первое, чем мы займемся в этой работе – избавимся от «вредной» конструкции

**ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры.**

В режиме Конфигуратор

Откройте модуль документа **ОказаниеУслуги**. Из процедуры обработки проведения видно, что все данные, необходимые для проведения документа, мы получаем из самого документа и только для определения типа номенклатуры (товар или услуга) – читая данные всего объекта Номенклатура.

Обращение к объекту **Номенклатура**:

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры

Это не единственные данные, которые содержатся не в самом документе и которые будут нужны для его правильного проведения.

Для оптимизации поступим следующим образом: все данные, связанные с номенклатурой, которая содержится в табличной части документа, мы будем получать с помощью запроса к БД. А данные, связанные с самим документом (дата, склад..) будем по-прежнему получать из документа. Такой подход позволит читать только нужные данные и максимально ускорить проведение документа.

Запросом мы будем получать:

- Номенклатуру
- Количество
- Сумму
- Стоимость

Из документа возьмем:

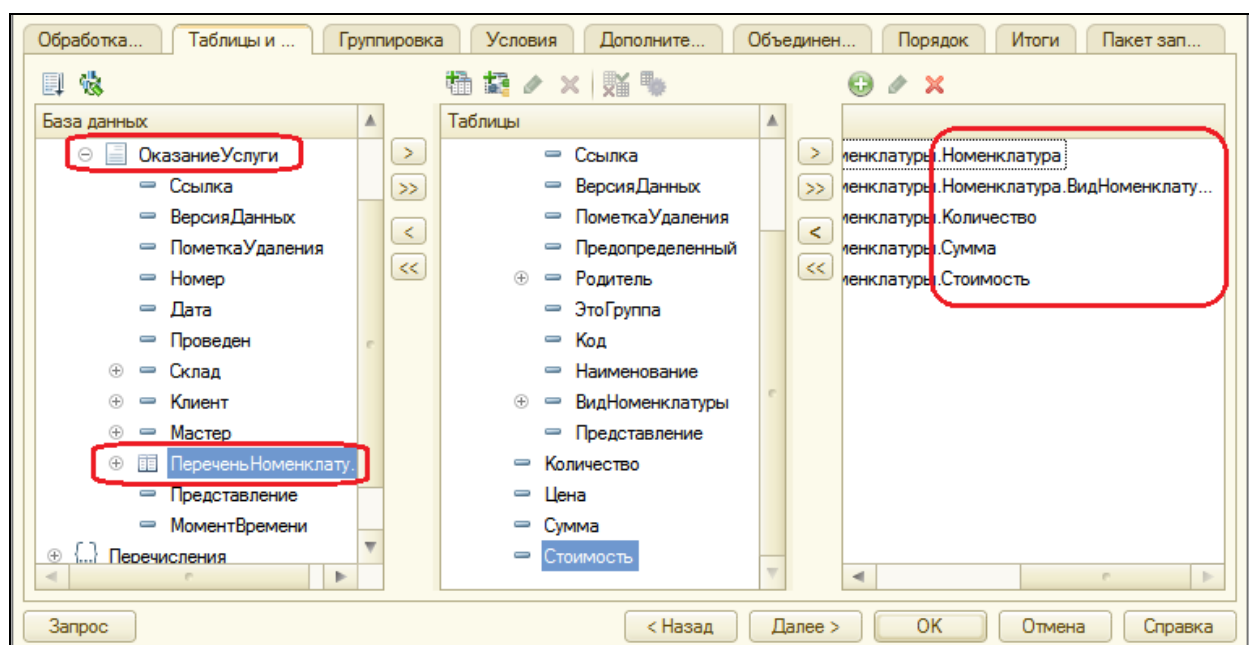
- Дата

- Клиент
- Мастер
- Склад

Установите курсор перед циклом Если... и из контекстного меню выберите **Конструктор запроса с обработкой результата**. Подтвердите создание нового запроса.

В окне конструктора перейдите на вкладку **Таблицы и поля** и выберите таблицу **ОказаниеУслугиПереченьНоменклатуры** – это табличная часть документа **ОказаниеУслуги**.

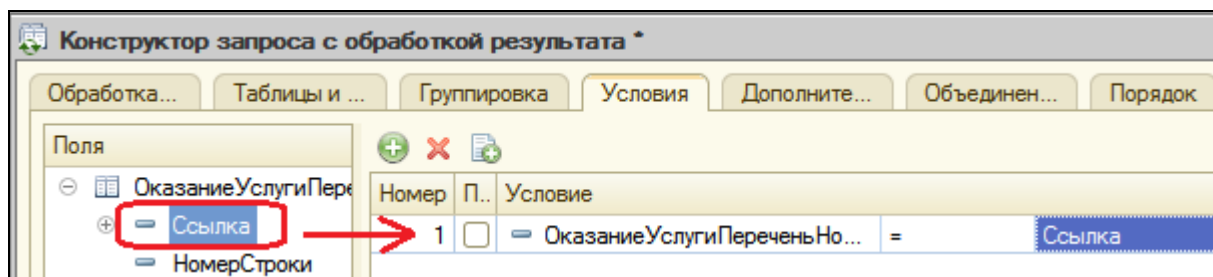
Из этой таблицы нам нужны поля – **Номенклатура, Номенклатура.ВидНоменклатуры, Количество, Сумма и Стоимость**.



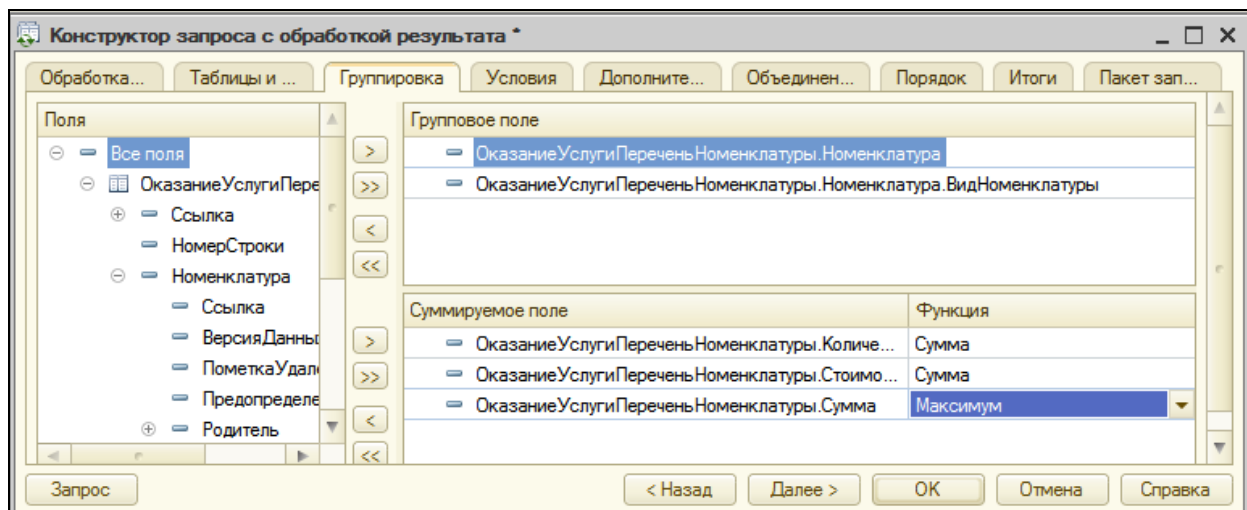
Но нам нужны не все записи этой таблицы, а только те, которые относятся к нашему документу. Поэтому перейдите на вкладку **Условия** и задайте условие отбора из таблицы только строк проводимого документа. Для этого перетащите поле **Ссылка** в список условий запроса:

В обработчике появится условие:

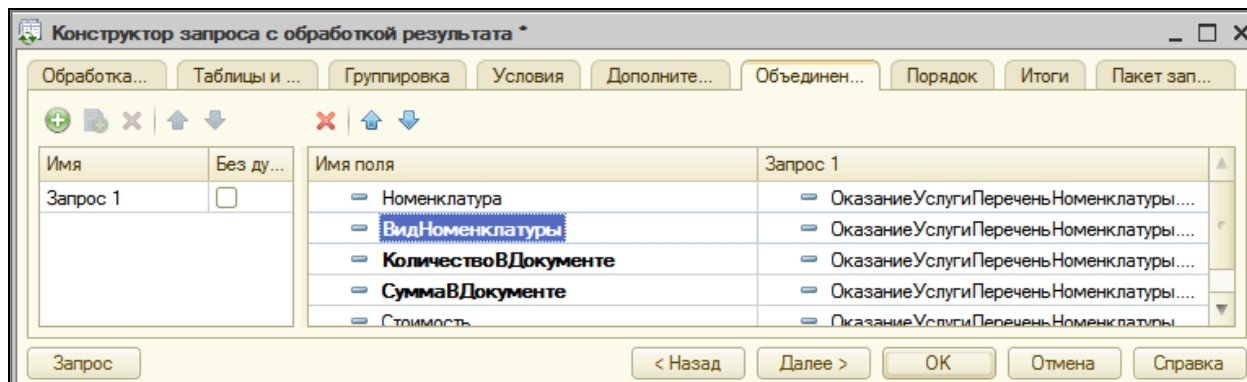
ОказаниеУслугиПереченьНоменклатуры.Ссылка=&Ссылка



Следует учесть, что в табличной части документа одна и та же номенклатура может встречаться несколько раз. Поэтому на закладке **Группировка** сгруппируйте записи по полю **Номенклатура** и **НоменклатураВидНоменклатуры**, а рассчитывать будем сумму значений для полей **Количество** и **Сумма**. Также в состав суммируемых полей включим поле **Стоимость**. По нему будем рассчитывать, например, функцию **Максимум**.



На закладке **Объединения/Псевдонимы** задайте псевдонимы для полей **Количество** и **Сумма** – **КоличествоВДокументе** и **СуммаВДокументе**, а для поля **НоменклатураВидНоменклатуры** – **ВидНоменклатуры**.



Нажмите OK и посмотрите какой текст запроса сформирован.

```

//{{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения
будут утеряны!!!

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
          |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
          |
          |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры      КАК
ВидНоменклатуры,
          |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество)      КАК
КоличествоВДокументе,
          |      МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Сумма)      КАК
СуммаВДокументе,
          |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Стоимость)      КАК
Стоимость
          |ИЗ
          |      Документ.ОказаниеУслуги.ПереченьНоменклатуры      КАК
ОказаниеУслугиПереченьНоменклатуры
          |ГДЕ
          |      ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
          |
          |СГРУППИРОВАТЬ ПО
          |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
          |
          |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

    Запрос.УстановитьПараметр("Ссылка", Ссылка);

    Результат = Запрос.Выполнить();

    ВыборкаДетальныеЗаписи = Результат.Выбрать();

    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
        // Вставить обработку выборки ВыборкаДетальныеЗаписи
    КонечЦикла;

//{{}}КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

```

Проанализируем текст. Для работы с запросами используется объект встроенного языка **Запрос**. Вначале создается новый объект **Запрос** и помещается в переменную **Запрос**. Затем в свойство **Текст** объекта **Запрос** помещается сам текст запроса (**Запрос.Текст=...**). После этого устанавливается значение параметра запроса **&Ссылка** как ссылка на тот документ, в модуле которого мы сейчас находимся.

```
Запрос.УстановитьПараметр("Ссылка", Ссылка);
```

Затем запросы выполняется (**Запрос.Выполнить()**), получается объект **РезультатЗапроса**, и выполняется его метод **Выбрать()**, который формирует выборку записей из результата запроса.

Получается объект **ВыборкаИзРезультатаЗапроса**, который помещается в переменную **ВыборкаДетальныеЗаписи**.

Далее, используя метод этого объекта **Следующий()** (**ВыборкаДетальныеЗаписи.Следующий()**), мы будем в цикле обходить выборку записей запроса. Выполняя метод выборки запроса **ВыборкаДетальныеЗаписи.Следующий()**, мы на каждом шаге цикла позиционируем указатель на следующую запись выборки, пока не будет достигнут конец выборки.

Чтобы в цикле получить значение какого-либо поля выборки из результата запроса, мы будем обращаться к полям запроса через точку от переменной **ВыборкаДетальныеЗаписи**, которая содержит текущую строку выборки запроса. Например, так: **ВыборкаДетальныеЗаписи.Номенклатура**.

Теперь осталось перенести существовавшие в этом модуле ранее строки, описывающие движения регистров, внутрь цикла обхода результата запроса.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    // Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;
```

Новый текст в листинге будет выделен **жирным** текстом для удобства восприятия.

Сначала вместо комментария «// Вставить обработку выборки ВыборкаДетальныеЗаписи» перенесите условие проверки и весь код, формирующий движения по регистрам **ОстаткиМатериалов** и **СтоимостьМатериалов**.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если
ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал =
ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Склад = Склад;
        Движение.Количество =
ТекСтрокаПереченьНоменклатуры.Количество;

        // регистр СтоимостьМатериалов Расход
        Движение = Движения.СтоимостьМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
```

```

        Движение.Материал
ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Стоимость
ТекСтрокаПереченьНоменклатуры.Количество*ТекСтрокаПереченьНоменклатуры.Сто
мость;
        КонецЕсли;

    КонецЦикла;

```

В условии замените ТекСтрокаПереченьНоменклатуры.Номенклатура на ВыборкаДетальныеЗаписи, так как вид номенклатуры мы теперь получаем из запроса. В движениях также заменим ТекСтрокаПереченьНоменклатуры на ВыборкаДетальныеЗаписи. Для поля **Количество** мы задали псевдоним в запросе, поэтому заменим его на **КоличествоВДокументе**.

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.ВидНоменклатуры
Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Склад = Склад;
        Движение.Количество
        ВыборкаДетальныеЗаписи.КоличествоВДокументе;

        // регистр СтоимостьМатериалов Расход
        Движение = Движения.СтоимостьМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Стоимость
        ВыборкаДетальныеЗаписи.КоличествоВДокументе*ВыборкаДетальныеЗаписи.Сто
имость;
        КонецЕсли;

    КонецЦикла;

```

Теперь перенесем формирование движений по регистру **Продажи**:

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.ВидНоменклатуры
Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Склад = Склад;

```

```

        Движение.Количество
ВыборкаДетальныеЗаписи.КоличествоВДокументе;

        // регистр СтоимостьМатериалов Расход
        Движение = Движения.СтоимостьМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Стоимость
ВыборкаДетальныеЗаписи.КоличествоВДокументе*ВыборкаДетальныеЗаписи.Стоимость
;
        КонецЕсли;
    // Регистр Продажи
        Движение = Движения.Продажи.Добавить();
        Движение.Период = Дата;
        Движение.Номенклатура
ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Клиент = Клиент;
        Движение.Мастер = Мастер;
        Движение.Количество
ТекСтрокаПереченьНоменклатуры.Количество;
        Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
        Движение.Стоимость
ТекСтрокаПереченьНоменклатуры.Стоимость
ТекСтрокаПереченьНоменклатуры.Количество;
КонецЦикла;

```

Здесь произведем аналогичные замены. ТекСтрокаПереченьНоменклатуры заменим на ВыборкаДетальныеЗаписи. А также поля Сумма и Количество заменим на их псевдонимы СуммаВДокументе и КоличествоВДокументе.

```

// Регистр Продажи
        Движение = Движения.Продажи.Добавить();
        Движение.Период = Дата;
        Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Клиент = Клиент;
        Движение.Мастер = Мастер;
        Движение.Количество
ВыборкаДетальныеЗаписи.КоличествоВДокументе;
        Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
        Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость
ВыборкаДетальныеЗаписи.КоличествоВДокументе;
        КонецЦикла;

```

Оставшийся цикл обхода табличной части можно удалить:

```

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл

        КонецЦикла;

```



## Процедура проведения примет следующий вид:

Процедура ОбработкаПроведения(Отказ, Режим)

```

Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Движения.Продажи.Записывать = Истина;

    //{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

Запрос = Новый Запрос;
Запрос.Текст =
    "ВЫБРАТЬ
      | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
      | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры      КАК
ВидНоменклатуры,
      | СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество)                  КАК
КоличествоВДокументе,
      | СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,
      | МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость
      | ИЗ
      | Документ.ОказаниеУслуги.ПереченьНоменклатуры                        КАК
ОказаниеУслугиПереченьНоменклатуры
      | ГДЕ
      | ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
      |
      | СГРУППИРОВАТЬ ПО
      | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
      | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Результат = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Склад = Склад;
        Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

        // регистр СтоимостьМатериалов Расход
        Движение = Движения.СтоимостьМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Стоимость =
        ВыборкаДетальныеЗаписи.КоличествоВДокументе*ВыборкаДетальныеЗаписи.Стоимость;
        КонецЕсли;
    // Регистр Продажи
    Движение = Движения.Продажи.Добавить();
    Движение.Период = Дата;
    Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

```

```

Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость *
ВыборкаДетальныеЗаписи.КоличествоВДокументе;

КонецЦикла;

КонецПроцедуры

```

Теперь запустите 1С: Предприятие в режиме отладки и перепроведите документы **ОказаниеУслуги**, проверьте, что ничего не изменилось.

Т.о. мы выполнили первый пункт нашего плана – избавились в процедуре проведения от считывания всех данных объекта **Номенклатура**, оптимизировав выполнение процедуры проведения.

### Автоматический расчет стоимости

Приступим ко второму этапу плана. До сих пор стоимость расходующихся материалов мы вписывали в документ **Оказание услуги** вручную, при его создании.

Теперь же будем определять стоимость номенклатуры «по среднему»: для каждой номенклатуры делить ее общую, суммарную стоимость на имеющееся у нас количество номенклатуры, т.о. получая среднюю стоимость единицы номенклатуры.

Чтобы выполнить такой расчет, понадобятся дополнительные данные, которых сейчас нет. Для каждой номенклатуры из табличной части нам понадобятся:

- Ее стоимость, хранящаяся в регистре **СтоимостьМатериалов**;
- Общее ее количество на всех складах, хранящееся в регистре **ОстаткиМатериалов**.

Поэтому нам нужно будет доработать запрос, чтобы он получал из БД и эти данные тоже. Т.о. нам хотелось бы, чтобы запрос возвращал следующие поля для каждой номенклатуры, которая есть в документе:

| Табличная часть документа |                       |                      |                     | Регистр<br>Стоимость<br>материалов | Регистр<br>Остатки<br>материалов |
|---------------------------|-----------------------|----------------------|---------------------|------------------------------------|----------------------------------|
| Номенклатура              | Кол-во в<br>документе | Сумма в<br>документе | Вид<br>номенклатуры | Стоимость                          | Кол-во на всех<br>складах        |

Первые четыре поля мы уже получаем из табличной части самого документа, а последние два нужно будет получить из других таблиц.

Это значит, что наш запрос должен содержать два левых соединения таблицы документа с другими таблицами: одно – с таблицей **РегистрНакопления.СтоимостьМатериалов.Остатки**; другое – с таблицей **РегистрНакопления.ОстаткиМатериалов.Остатки**.

Важная деталь: в предложенной схеме виртуальные таблицы будут возвращать стоимость и остатки абсолютно для всей номенклатуры, когда нас интересует только та, которая указана в нашем документе.

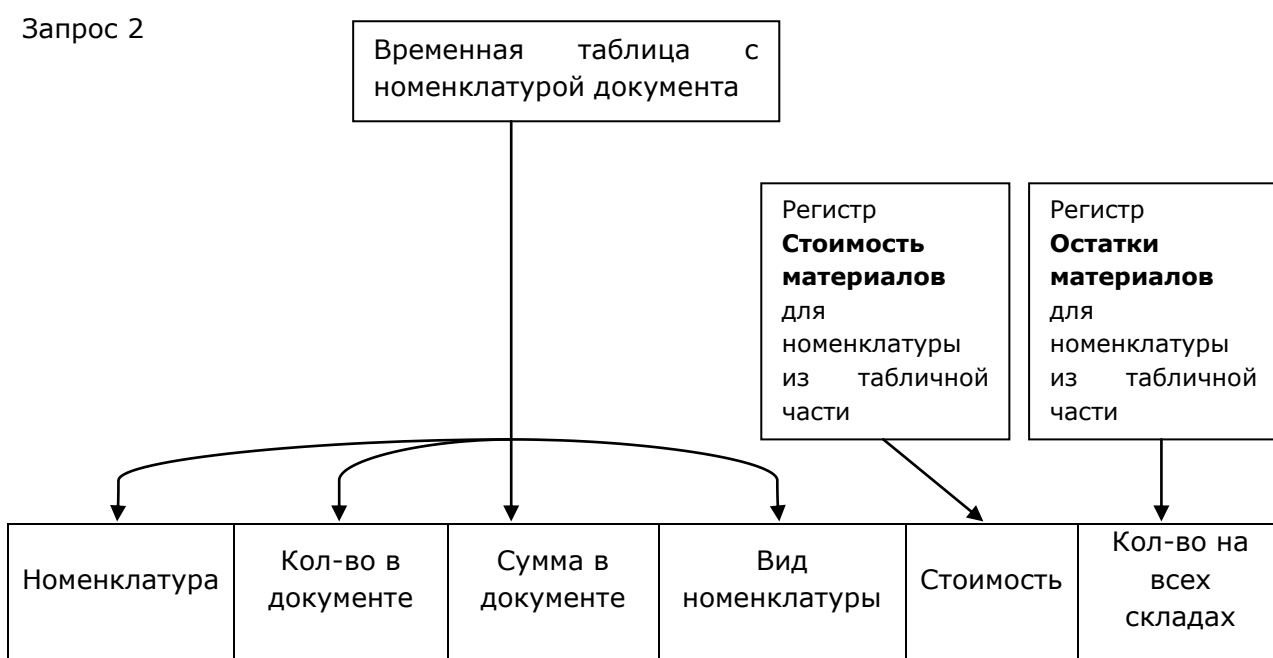
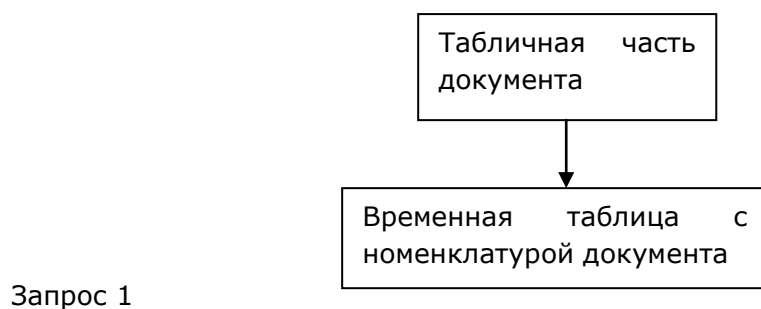
Для маленькой базы это не важно, но для реальной БД, где 15 000 наименований и только 5 используются в документе – это непозволительная расточительность вычислительных ресурсов.

Поэтому во все виртуальные таблицы, которые мы будем использовать, нужно добавить условие отбора только номенклатуры из табличной части нашего документа. В этом случае стоимость и остатки будут рассчитаны не для всей номенклатуры вообще, а только для нужной нам.

Чтобы не получать список номенклатуры три раза (для документа и в каждой виртуальной таблице заново), мы можем сформировать его заранее и затем уже использовать в нужных нам условиях запроса.

Выполнить эту задачу нам помогут *временные таблицы* – программные объекты, которые разработчик может создать и заполнить данными, а запросы могут использовать данные временных таблиц для своих нужд.

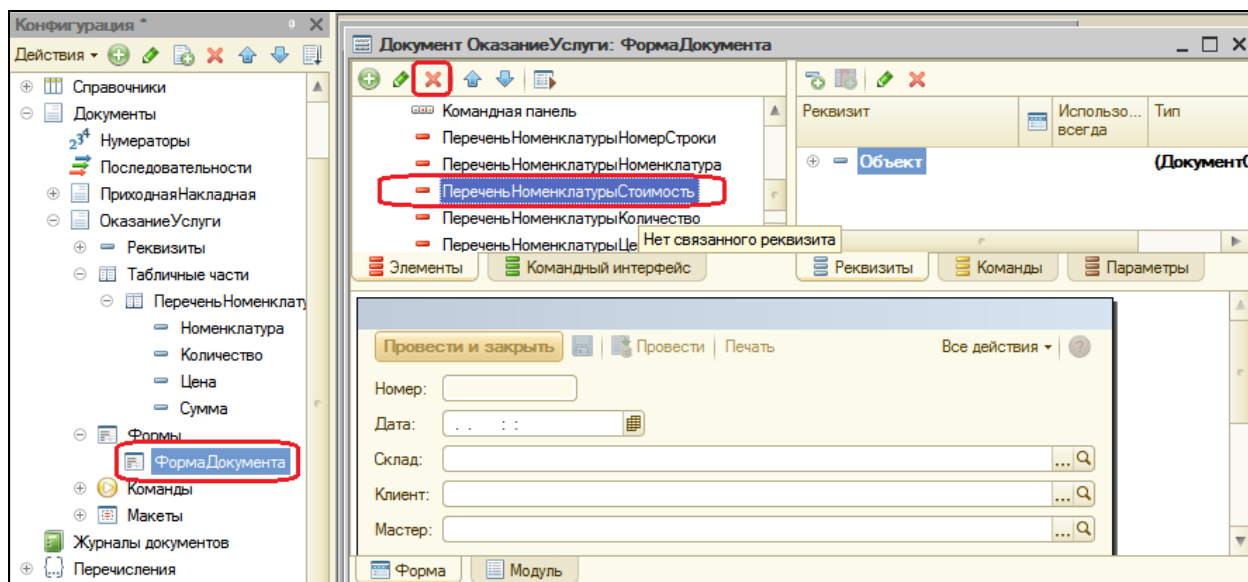
Таким образом, схема нашего запроса приобретает следующий вид:



## В режиме Конфигуратор

Первое – мы удалим реквизит табличной части **Стоимость** документа **ОказаниеУслуги**, который нам больше не понадобится. Для этого откройте окно редактирования объекта Документа **ОказаниеУслуги**, перейдите на закладку **Данные**, раскройте список реквизитов табличной части документа, выделите реквизит **Стоимость** и нажмите кнопку **Удалить** в командной панели. Хотя проще это сделать из окна конфигурации.

Также удалите поле **Стоимость** из таблицы **ПереченьНоменклатуры**, расположенной в форме.



Временную таблицу мы сформируем с помощью того запроса, который у нас уже написан. Откройте модуль документа **ОказаниеУслуги**.

В процедуре **ОбработкаПроведения()** перед созданием запроса создайте менеджер временных таблиц и укажите, что этот запрос будет использовать созданный менеджер:

Процедура ОбработкаПроведения(Отказ, Режим)

```
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Движения.Продажи.Записывать = Истина;
```

```
//Создать менеджер временных таблиц.
МенеджерВТ = Новый МенеджерВременныхТаблиц;
```

```
Запрос = Новый Запрос;
//Укажем, какой менеджер временных таблиц использует этот запрос
Запрос.МенеджерВременныхТаблиц = МенеджерВТ;
```

```
Запрос.Текст =
    "ВЫБРАТЬ
    | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
```

Теперь изменим запрос, чтобы он создавал временную таблицу, которая будет храниться в нашем менеджере временных таблиц **МенеджерВТ**.

Чтобы конструктор запроса смог открыть наш запрос, удалите из него строку (поля Стоимость у нас больше нет) и запятую в строке выше этой:

МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость

Получится следующее:

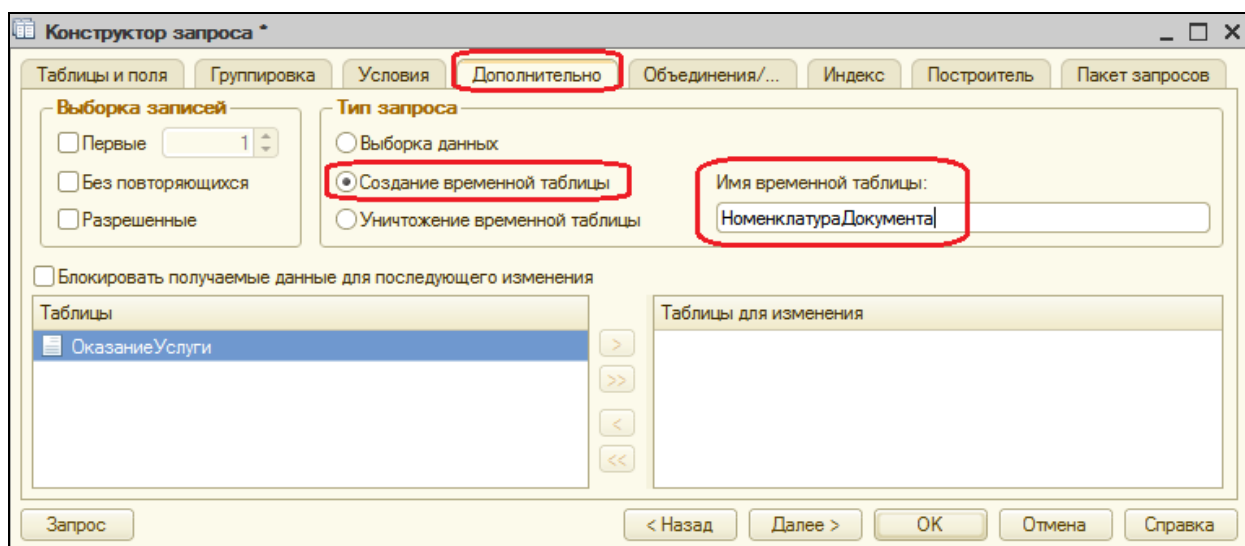
```

Запрос.Текст =
    "ВЫБРАТЬ
      |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
      |
      |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры      КАК
ВидНоменклатуры,
      |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество)      КАК
КоличествоВДокументе,
      |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма)      КАК
СуммаВДокументе
      |
      |ИЗ

```

Установите курсор внутрь текста запроса (например, на слове **ВЫБРАТЬ**) и выполните команду контекстного меню **Конструктор запроса**.

Чтобы результат запроса поместить во временную таблицу, перейдите на закладку **Дополнительно** и отметьте пункт **Создание временной таблицы**. Задайте ей имя – **НоменклатураДокумента**. Нажмите ОК.



Конструктор сформировал текст запроса с одной новой строкой:

```

"ВЫБРАТЬ
      |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
      |
      |      ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры      КАК
ВидНоменклатуры,
      |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество)      КАК
КоличествоВДокументе,
      |      СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма)      КАК
СуммаВДокументе
      | ПОМЕСТИТЬ НоменклатураДокумента
      |ИЗ
      |      Документ.ОказаниеУслуги.ПереченьНоменклатуры      КАК
ОказаниеУслугиПереченьНоменклатуры

```

```

|ГДЕ
|    ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
|
|СГРУППИРОВАТЬ ПО
|    ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
|
|ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

```

Это означает, что результат запроса будет сохранен во временной таблице **НоменклатураДокумента**. Это был Запрос 1 на нашей схеме.

Теперь если мы для другого запроса укажем этот же самый менеджер временных таблиц МенеджерВТ, то в другом запросе мы сможем обратиться к данным этой временной таблицы.

Займемся вторым запросом.

Установите курсор на следующую строку после оператора Результат = Запрос.Выполнить(); (именно здесь выполняется создание временной таблицы) и напишите заготовку будущего запроса:


```

Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос2.Текст = "";

```

Мы создали новый объект Запрос и назначили ему тот же самый менеджер временных таблиц, чтобы иметь возможность обращаться к созданной нами ранее временной таблице.

Установите курсор внутри кавычек и выполните контекстную команду **Конструктор запроса**. Согласитесь на создание нового запроса.

Создадим описание временной таблицы в этом запросе. Для этого над списком Таблицы нажмите **Создать описание временной таблицы** .

Введите имя нашей временной таблицы **НоменклатураДокумента** и добавьте поля:

- Номенклатура, тип СправочникСсылка.Номенклатура;
- ВидНоменклатуры, тип ПеречислениеСсылка.ВидыНоменклатуры;
- КоличествоВДокументе, тип Число, 15, 3;
- СуммаВДокументе, тип Число, 15, 2.

Нажмите ОК.

**Временная таблица**

Имя таблицы:

Поля:

| Имя поля             | Тип значения              |
|----------------------|---------------------------|
| Номенклатура         | СправочникСсылка.Номен... |
| ВидНоменклатуры      | ПеречислениеСсылка.Вид... |
| КоличествоВДокументе | Число                     |
| СуммаВДокументе      | Число                     |

OK Отмена Справка

Выберите из **НоменклатураДокумента** все поля и нажмите кнопку **Запрос**.

**Конструктор запроса \***

Таблицы и ... Группировка Условия Дополните... Объединен... Порядок Итоги Построитель Пакет зап...

База данных

- Справочники
- Документы
- ПриходнаяНакладная
- ОказаниеУслуги
- Перечисления
- РегистрыСведений
- РегистрыНакопления

Таблицы

- НоменклатураДокумента
- Номенклатура
- ВидНоменклатуры
- Ссылка
- Порядок
- КоличествоВДокументе
- СуммаВДокументе

Поля

- НоменклатураДокумента.Номенклат
- НоменклатураДокумента.ВидНоменк
- НоменклатураДокумента.КоличествоВДокументе
- НоменклатураДокумента.СуммаВДокументе

Запрос

< Назад Далее > OK Отмена Справка

**ВЫБРАТЬ**

НоменклатураДокумента.Номенклатура,  
НоменклатураДокумента.ВидНоменклатуры,  
НоменклатураДокумента.КоличествоВДокументе,  
НоменклатураДокумента.СуммаВДокументе

**ИЗ**

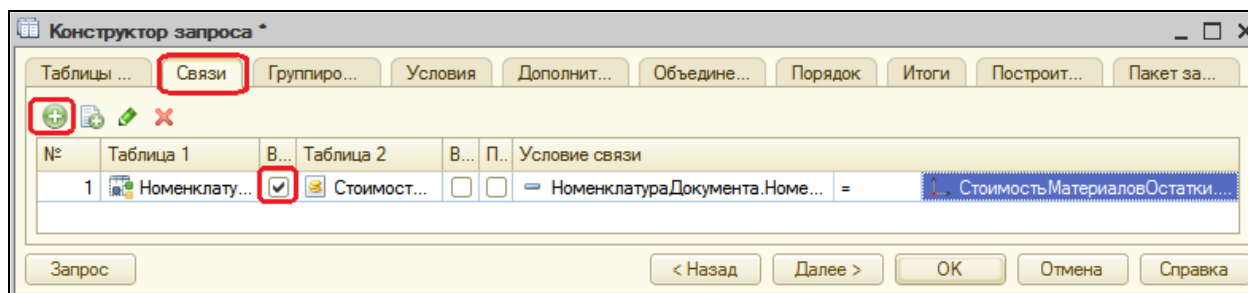
НоменклатураДокумента КАК НоменклатураДокумента


Итак, мы создали первую часть второго запроса – выбрали информацию из временной таблицы. Теперь будем соединять эту конструкцию левыми соединениями с таблицами остатков.

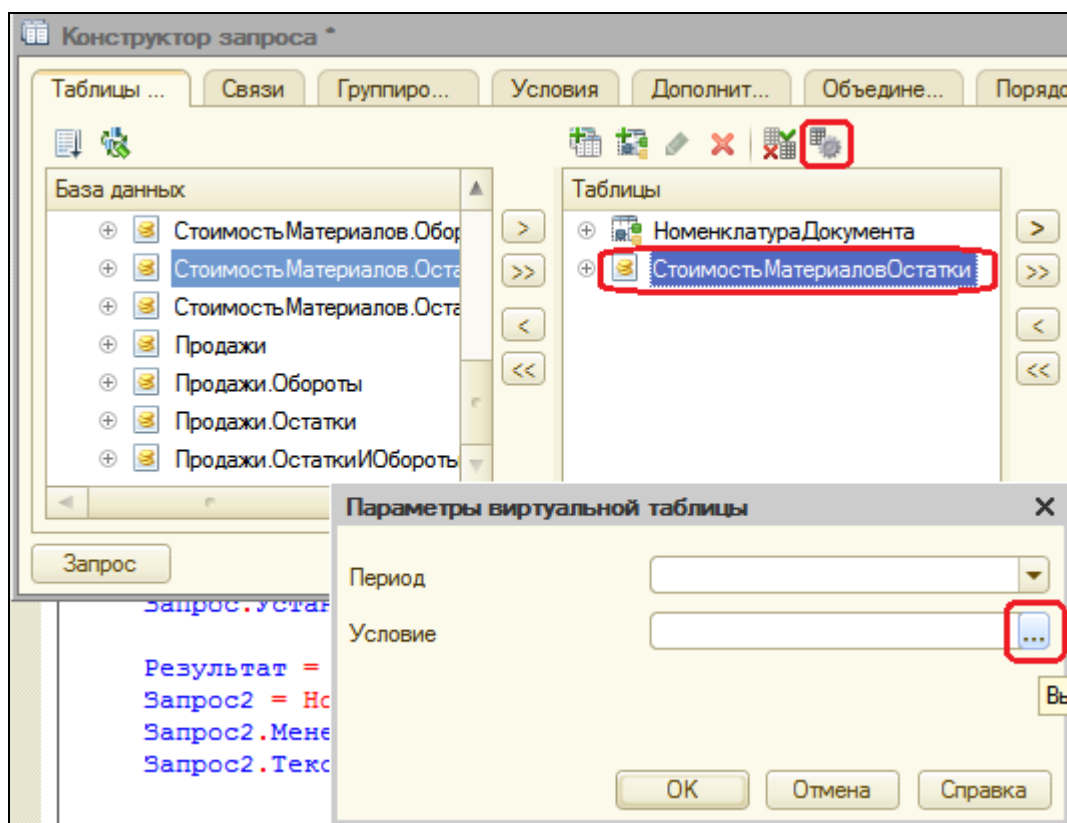
Добавим в список таблиц запроса виртуальную таблицу **РегистрНакопления.СтоимостьМатериалов.Остатки**. Из нее выберем поле **СтоимостьОстаток**. Перейдем на закладку **Связи** и зададим связь между таблицами.



Из временной таблицы будем выбирать все записи, и поле **Номенклатура** временной таблицы должно быть равно полю **Материал** таблицы остатков.



Ограничим виртуальную таблицу только нужной номенклатурой. Для этого вернемся на вкладку таблицы и поля, выделите в списке таблицу **СтоимостьМатериаловОстатки** и нажмите кнопку  **Параметры виртуальной таблицы**.



Задайте условие:

|                        |   |          |                                    |    |
|------------------------|---|----------|------------------------------------|----|
| Материал               | В | (ВЫБРАТЬ | НоменклатураДокумента.Номенклатура | ИЗ |
| НоменклатураДокумента) |   |          |                                    |    |

Т.е. материал должен быть среди номенклатуры, выбранной из временной таблицы. Нажмите ОК.

Теперь нажмите кнопку **Запрос** и посмотрите на текст, сформированный конструктором.

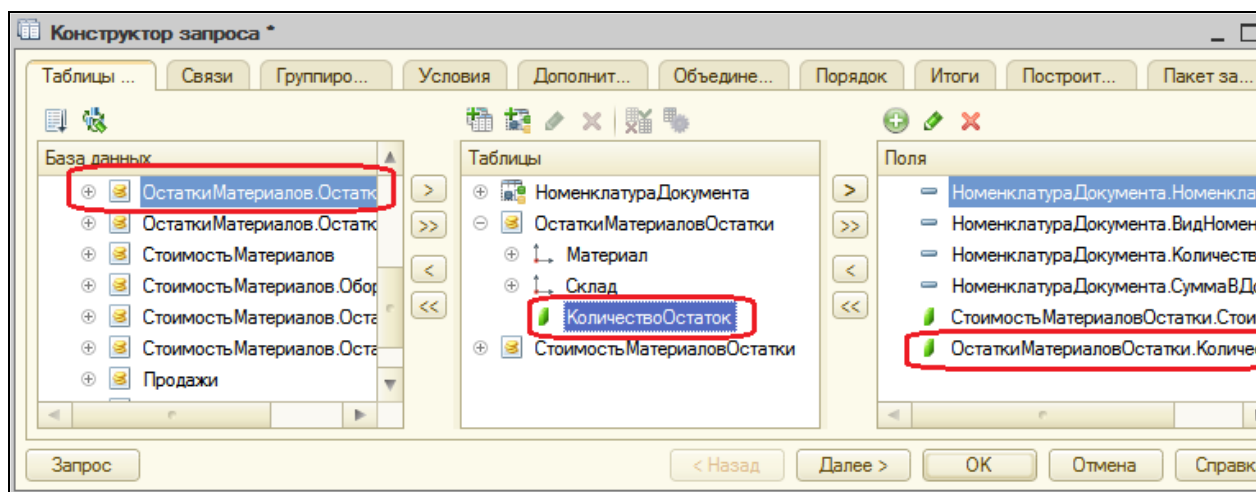
```

ВЫБРАТЬ
    НоменклатураДокумента.Номенклатура,
    НоменклатураДокумента.ВидНоменклатуры,
    НоменклатураДокумента.КоличествоВДокументе,
    НоменклатураДокумента.СуммаВДокументе,
    СтоимостьМатериаловОстатки.СтоимостьОстаток
ИЗ
    НоменклатураДокумента КАК НоменклатураДокумента
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.Остатки(
        '
        Материал В
        (ВЫБРАТЬ
            НоменклатураДокумента.Номенклатура
        ИЗ
            НоменклатураДокумента))
        КАК
        СтоимостьМатериаловОстатки
        ПО
            НоменклатураДокумента.Номенклатура
        СтоимостьМатериаловОстатки.Материал
    =

```

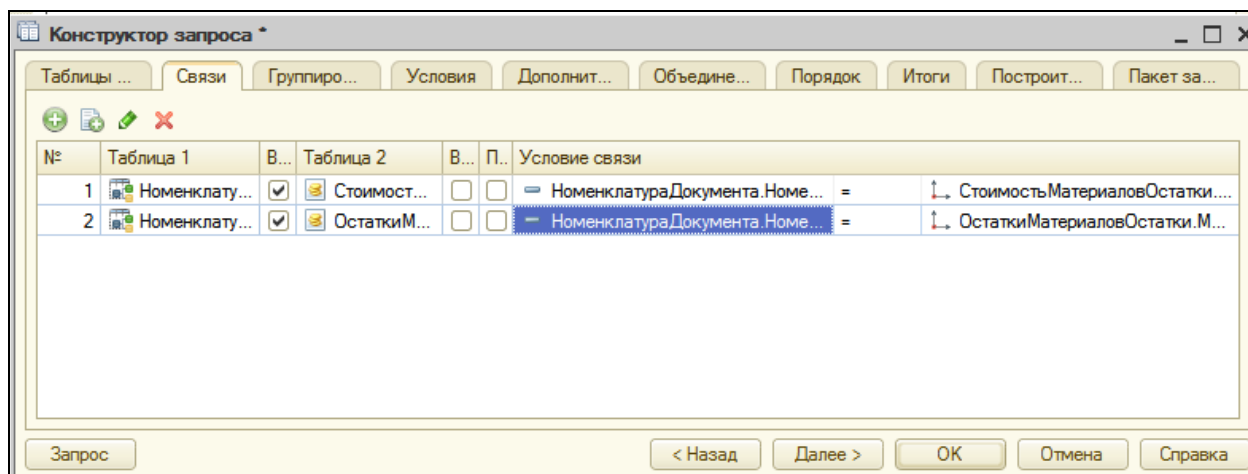
Тем самым мы добавили к выбранным ранее полям стоимость номенклатуры.

Теперь добавим виртуальную таблицу остатков регистра **ОстаткиМатериалов.Остатки**, из которой выберем поле **Количество.Остаток**.



Перейдите на закладку **Связи** и задайте связь между таблицами.

Из временной таблицы будем выбирать все записи, поле **Номенклатура** временной таблицы должно быть равно полю **Материал** таблицы остатков.



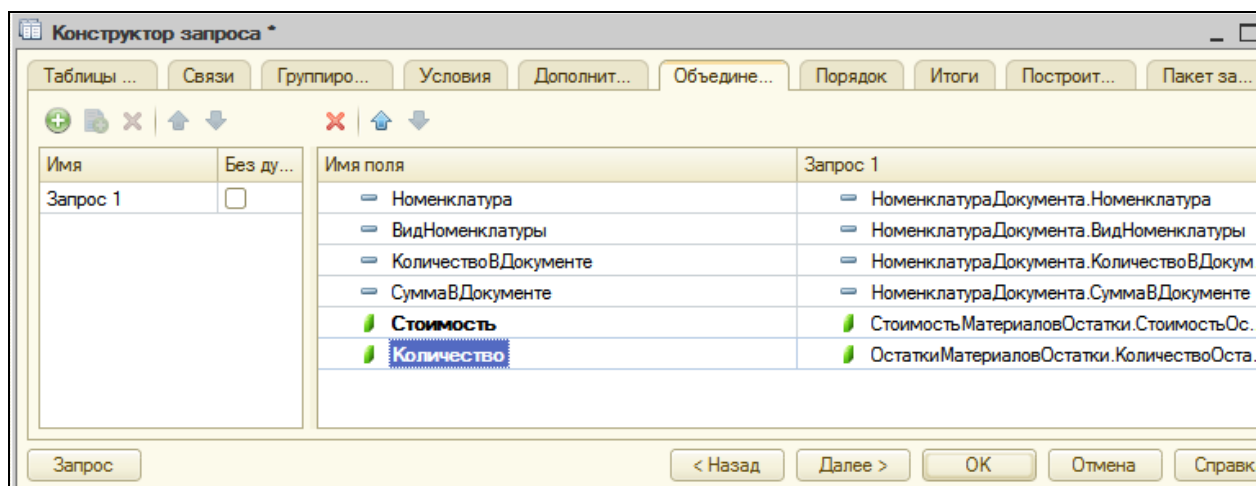
Также зададим параметры виртуальной таблицы **ОстаткиМатериаловОстатки**. В параметр Условие введите:

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)

Тем самым мы добавили к выбранным ранее полям остатки номенклатуры на всех складах.

В заключение перейдем на закладку **Объединения/Псевдонимы** и зададим следующие псевдонимы полей:

- **СтоимостьОстаток – Стоимость;**
- **КоличествоОстаток – Количество.**



В результате получим следующий текст запроса:

```


ВЫБРАТЬ
    НоменклатураДокумента.Номенклатура,
    НоменклатураДокумента.ВидНоменклатуры,
    НоменклатураДокумента.КоличествоВДокументе,
    НоменклатураДокумента.СуммаВДокументе,
    СтоимостьМатериаловОстатки.СтоимостьОстаток КАК Стоимость,
    ОстаткиМатериаловОстатки.КоличествоОстаток КАК Количество
ИЗ
    НоменклатураДокумента КАК НоменклатураДокумента
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.Остатки(
            '
            Материал В
            (ВЫБРАТЬ
                НоменклатураДокумента.Номенклатура
            ИЗ
                НоменклатураДокумента))
            КАК
            СтоимостьМатериаловОстатки
            ПО
                НоменклатураДокумента.Номенклатура
            =
            СтоимостьМатериаловОстатки.Материал
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки(
            '
            Материал В
            (ВЫБРАТЬ
                НоменклатураДокумента.Номенклатура
            ИЗ
                НоменклатураДокумента))
            КАК
            ОстаткиМатериаловОстатки
            ПО
                НоменклатураДокумента.Номенклатура
            =
            ОстаткиМатериаловОстатки.Материал

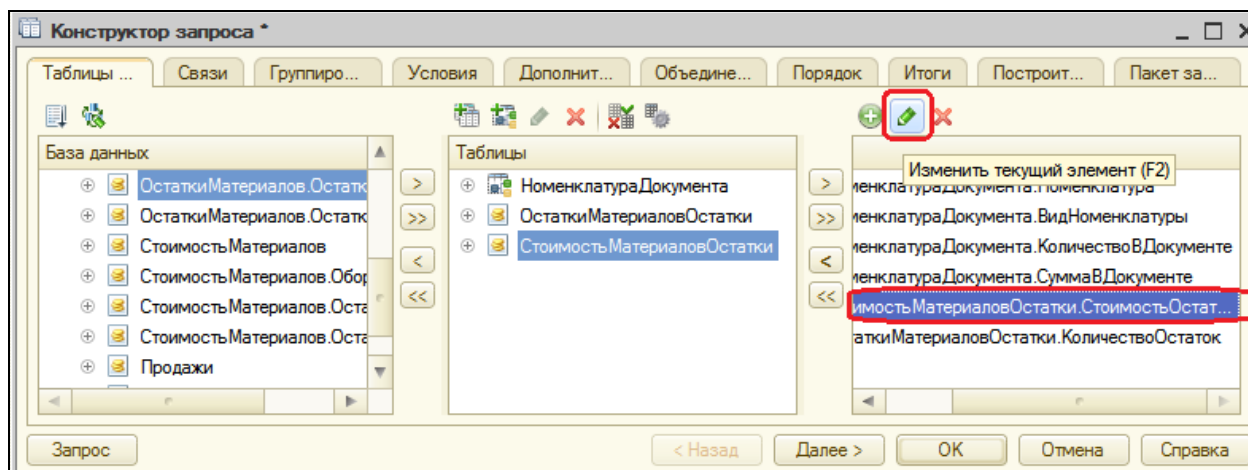
```

Нужно предусмотреть тот случай, когда номенклатура в справочнике есть, но у нее нет ни остатков, ни стоимости. Это может быть, например, когда номенклатуру создали в справочнике, но она еще не поступала в нашу фирму.

В такой ситуации левые соединения с виртуальными таблицами не вернут ничего. На языке запросов это значит, что в полях **Стоимость** и **Количество** будут значения **NULL**.

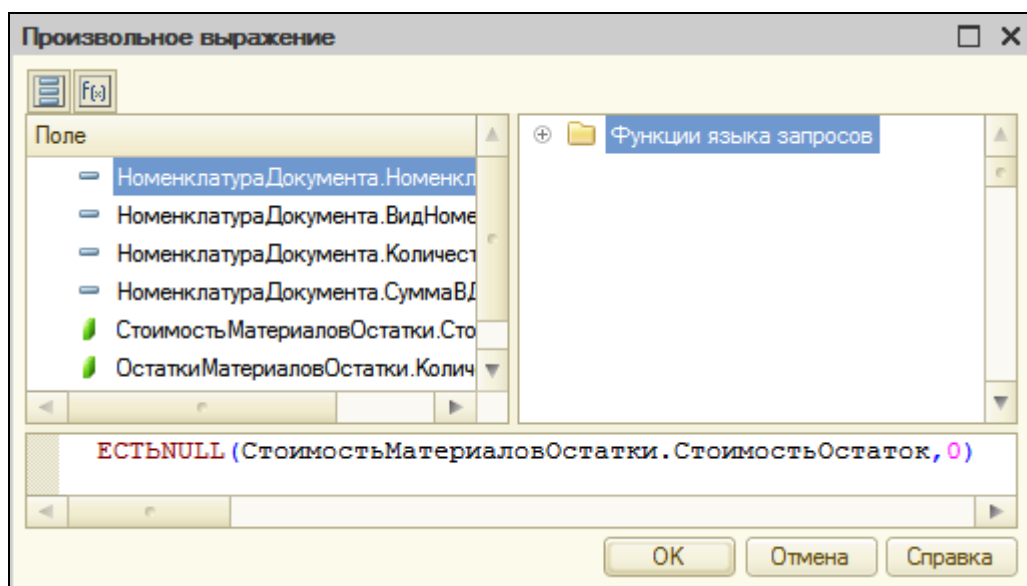
Избавимся в самом запросе от этих значений. Для этого мы применим функцию **ЕСТЬNULL()** к полям **Стоимость** и **Количество**. Если значение этого поля будет **NULL**, функция вернет **0**. В остальных случаях функция вернет само значение этого поля.

Перейдите на закладку **Таблицы и поля**, выделите **СтоимостьМатериаловОстатки.СтоимостьОстаток** и нажмите кнопку  **Изменить текущий элемент**.



Отредактируем значение поля:

ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток,0)



Аналогично с другим полем – **ОстаткиМатериаловОстатки.КоличествоОстаток.**

ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток,0)

Нажмите OK – текст запроса будет вставлен в модуль. Останется всего лишь дописать после него оператор выполнения запроса:

```
Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос2.Текст = "ВЫБРАТЬ
    | НоменклатураДокумента.Номенклатура,
    | НоменклатураДокумента.ВидНоменклатуры,
    | НоменклатураДокумента.КоличествоВДокументе,
    | НоменклатураДокумента.СуммаВДокументе,
    | ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0) КАК
Стоимость,
```

```

Количество | ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток, 0) КАК
            | ИЗ
            | НоменклатураДокумента КАК НоменклатураДокумента
            | ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.СтоимостьМатериалов.Остатки(
            |
            | 'Материал В
            | (ВЫБРАТЬ
            |
            | НоменклатураДокумента.Номенклатура
            | ИЗ
            | НоменклатураДокумента)) КАК
СтоимостьМатериаловОстатки
            | ПО НоменклатураДокумента.Номенклатура =
СтоимостьМатериаловОстатки.Материал
            | ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ОстаткиМатериалов.Остатки(
            |
            | 'Материал В
            | (ВЫБРАТЬ
            |
            | НоменклатураДокумента.Номенклатура
            | ИЗ
            | НоменклатураДокумента)) КАК
ОстаткиМатериаловОстатки
            | ПО НоменклатураДокумента.Номенклатура =
ОстаткиМатериаловОстатки.Материал";
Результат = Запрос2.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

```

Теперь разберемся с записью движений. Все операторы, которые были написаны нами ранее, будут работать без изменений.

Единственное, что потребуется изменить – способ получения стоимости. Раньше мы просто брали ее из документа, теперь же нам нужно ее рассчитать на основании данных запроса. Стоимость материала равна частному от деления всей стоимости, полученной запросом (**Стоимость**) на общее количество материала на всех складах (**Количество**).

Но, как сказано ранее, **Количество** может быть равно 0, а на ноль делить нельзя. Поэтому сразу после начала цикла обхода результата запроса рассчитаем стоимость для текущей номенклатуры.

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.Количество = 0 Тогда
        СтоимостьМатериала = 0;
    Иначе
        СтоимостьМатериала =
ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
    КонецЕсли;

```

|  |  |   |
|--|--|---|
| Если                                   | ВыборкаДетальныеЗаписи.ВидНоменклатуры | = |
| Перечисления.ВидыНоменклатуры.Материал | Тогда                                  |   |

Теперь заменим расчет стоимости в движениях регистров **СтоимостьМатериалов** и **Продажи**.

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.Количество = 0 Тогда
        СтоимостьМатериала = 0;
    Иначе
        СтоимостьМатериала =
ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
        КонецЕсли;
        Если ВыборкаДетальныеЗаписи.ВидНоменклатуры
Перечисления.ВидыНоменклатуры.Материал Тогда

            // регистр ОстаткиМатериалов Расход
            Движение = Движения.ОстаткиМатериалов.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
            Движение.Склад = Склад;
            Движение.Количество =
ВыборкаДетальныеЗаписи.КоличествоВДокументе;

            // регистр СтоимостьМатериалов Расход
            Движение = Движения.СтоимостьМатериалов.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
            Движение.Стоимость =
ВыборкаДетальныеЗаписи.КоличествоВДокументе*СтоимостьМатериала;
        КонецЕсли;
    // Регистр Продажи
        Движение = Движения.Продажи.Добавить();
        Движение.Период = Дата;
        Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
        Движение.Клиент = Клиент;
        Движение.Мастер = Мастер;
        Движение.Количество =
ВыборкаДетальныеЗаписи.КоличествоВДокументе;
        Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
        Движение.Стоимость = СтоимостьМатериала *
ВыборкаДетальныеЗаписи.КоличествоВДокументе;

    КонецЦикла;

```

Теперь, если проводить этот документ в первый раз, результат получится правильный. Но, если документ был проведен ранее и мы заново решим его провести, получим неправильный результат. Дело в том, что когда мы находимся в обработчике проведения и этот документ был проведен ранее, то в БД существуют движения этого документа.

Таким образом, читая из базы стоимость и остатки материалов, мы прочитаем их с учетом движений документа. А это неправильно.

Чтобы в обработчике проведения документа прочитать данные без учета предыдущих движений, которые мог выполнять документ, нужно перед чтением записать пустые наборы записей в те регистры, из которых мы собираемся читать. В нашем случае это **СтоимостьМатериалов** и **ОстаткиМатериалов**.

Перед выполнением второго запроса добавим строки:

```
|      ПО      НоменклатураДокумента.Номенклатура      =  
ОстаткиМатериаловОстатки.Материал";  
  
//Запишем пустые наборы записей, чтобы читать остатки без  
учета данных в документе.  
Движения.СтоимостьМатериалов.Записать();  
Движения.ОстаткиМатериалов.Записать();  
  
Результат = Запрос2.Выполнить();
```

Запустите режим отладки и перепроведите все документы **Оказание услуги**, проверьте правильность занесения данных в регистры.

## Оперативное и неоперативное проведение документов

*Оперативное проведение* документов пользователями выполняется в режиме реального времени, т.е. отображает изменения, факты, свершающиеся в настоящее время. Оперативное проведение особенно актуально при многопользовательской работе. Поэтому при этом способе проведения следует осуществлять максимум проверок, способных исключить ошибки при вводе данных пользователями.

Например, при оперативном проведении следует выполнять контроль остатков на складе списываемой номенклатуры, чтобы исключить одновременную продажу одного товара несколькими продавцами.

Оперативность или неоперативность проведения документа определяется по его дате. Если дата проводимого документа совпадает с текущей, то система будет проводить такой документ в оперативном режиме, не задавая вопросов, и в обработке проведения об этом можно узнать, чтобы выстроить определенный алгоритм проведения.



Если дата документа меньше текущей, то такой документ система будет проводить в неоперативном режиме. При этом перед проведением она напомнит, что оперативно она его провести не может (вдруг пользователь ошибся). И если пользователь подтвердит, что хочет провести его неоперативно, то система проведет документ неоперативно.

*Неоперативное проведение* подразумевает отражение в базе данных фактов, которые свершились в прошлом или которые точно будут совершены в будущем. Поэтому задача неоперативного проведения документов – просто отразить в информационной базе данные о совершенных операциях.

При неоперативном проведении не имеет смысла производить целый ряд проверок, в частности – контроль остатков. Подразумевается, что если в процессе неоперативного проведения были допущены ошибки, то анализ полученного состояния БД является отдельной задачей, не относящейся к неоперативному проведению и выполняющейся не в момент проведения документа.

Если логика учета подразумевает, что какой-то документ должен проводиться будущей датой, для такого документа механизм оперативного проведения должен быть отключен в метаданных (на закладке **Движения** окна редактирования объекта конфигурации).

## Контроль остатков

Общая методика контроля остатков при проведении документа заключается в записи движения документа, а затем чтения из базы остатков. Если появились отрицательные остатки, значит такой документ проводить нельзя. Нужно сообщить пользователю каких материалов не хватает и отменить проведение документа. Нам осталось проконтролировать это.

## В режиме Конфигуратор

Сделаем заготовку. После цикла обхода результата запроса и перед концом процедуры напомним:

```
КонецЦикла;
```

```
    Движения.Записать();  
    Если Режим = РежимПроведенияДокумента.Оперативный Тогда  
        //проверить отрицательные остатки.
```

КонецЕсли;

КонецПроцедуры

Сначала мы записываем движения в регистры. Затем определяем режим проведения документа. При выполнении процедуры **ОбработкаПроведения()** вторым параметром (**Режим**) в нее передается режим проведения документа и значение этой переменной сравнивается со значением системного перечисления **РежимПроведенияДокумента**. В случае оперативного проведения мы будем выполнять контроль остатков.

Теперь сделаем заготовку запроса для проверки отрицательных остатков. Используем тот же менеджер виртуальных таблиц.

```
Если Режим = РежимПроведенияДокумента.Оперативный Тогда
    //проверить отрицательные остатки.
    Запрос3 = Новый Запрос;
    Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;
    Запрос3.Текст = "";
КонецЕсли;
```

Установите курсор внутри кавычек и вызовите конструктор запроса. Подтвердите создание нового запроса.

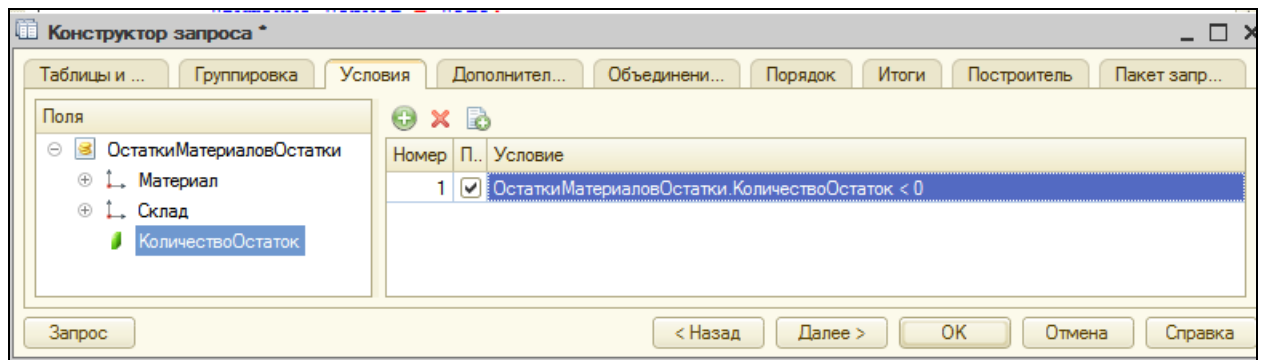
Выберите таблицу **ОстаткиМатериалов.Остатки** и из нее два поля: **Материал** и **КоличествоОстаток**. Задайте параметры этой таблице. В параметре **Условие** напишем:

```
Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ
НоменклатураДокумента) И Склад = &Склад
```

Т.е. мы получаем итоги только для той номенклатуры, которая содержится в нашей временной таблице и только по складу, который указан в документе.

Затем на вкладке **Условия** перенесем в список условий поле **ОстаткиМатериалов.Остатки.КоличествоОстаток**, поставим флажок **Произвольное** и укажем, что нас интересуют только отрицательные остатки:

```
ОстаткиМатериалов.Остатки.КоличествоОстаток < 0
```



Нажмите ОК.

Теперь осталось только установить параметр запроса, обойти результат запроса и вывести сообщения об отрицательных остатках.

```
//проверить отрицательные остатки.
Запрос3 = Новый Запрос;
Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос3.Текст = "ВЫБРАТЬ
    | ОстаткиМатериаловОстатки.Материал,
    | ОстаткиМатериаловОстатки.КоличествоОстаток
    |ИЗ
    | РегистрНакопления.ОстаткиМатериалов.Остатки(
    |
    |         Материал В
    |
    |         (ВЫБРАТЬ
    |
    |             НоменклатураДокумента.Номенклатура
    |
    |             ИЗ
    |                 НоменклатураДокумента)
    |
    |             И Склад = &Склад) КАК
    | ОстаткиМатериаловОстатки
    |ГДЕ
    | ОстаткиМатериаловОстатки.КоличествоОстаток < 0";
Запрос3.УстановитьПараметр("Склад", Склад);
Результат = Запрос3.Выполнить();
ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Сообщение = Новый СообщениеПользователю();
    Сообщение.Текст = "Не хватает" + Строка(-
    ВыборкаДетальныеЗаписи.КоличествоОстаток) + "единиц материала" +
    ВыборкаДетальныеЗаписи.Материал + " ";
    Сообщение.Сообщить();
    Отказ = Истина;
КонецЦикла;
КонецЕсли;

КонецПроцедуры
```

При выполнении проверки в запрос в параметре **Склад** передается склад, указанный в документе. Затем выполняется запрос для получения

отрицательных остатков номенклатуры, содержащейся во временной таблице и на складе, указанном в параметре **Склад**.

После этого выборка записей запроса обходится в цикле, и если есть такие записи, они выводятся в сообщениях пользователю.

При этом параметру **Отказ** присваивается значение **Истина**, т.е. документ не проводится, начатая транзакция отменяется, состояние данных, измененных в процессе проведения, возвращается в исходное, как до начала проведения документа.

### **Блокировка данных, которые читаются и изменяются при проведении**

Сейчас схема нашей процедуры такова:

1. Выполняем первый запрос с именем **Запрос**. В результате мы формируем временную таблицу из перечня номенклатуры документа.
2. Выполняем второй запрос с именем **Запрос2**. В результате мы читаем стоимость и остатки для номенклатуры, содержащейся в табличной части документа.
3. Записываем движения регистров (**Движения.Записать()**)
4. Выполняем третий запрос **Запрос3**. Тем самым мы проверяем наличие отрицательных остатков.

Обратите внимание, что, начиная с выполнения второго запроса и до конца процедуры, нам необходимо обеспечить неизменность стоимости и остатков номенклатуры, с которой мы работаем, и запретить другим транзакциям даже читать эти данные. Сама система заблокирует изменение этих данных, но лишь начиная с момента записи движения.

Однако может возникнуть следующая ситуация. Выполняя второй запрос, мы прочитали, что есть 2 шт. некоторого материала. И другая транзакция (другой пользователь), которая собирается списывать материалы, тоже прочитала, что есть 2 шт. этого материала. После этого мы записали движения, система заблокировала эти данные. Другая транзакция ждет, когда мы освободим данные. Мы провели документ, списали 2 шт. материала и освободили данные. Другая транзакция пытается тоже списать 2 шт. материала, но его уже нет!

Аналогичная ситуация может возникнуть и между пунктами 3 и 4, в результате чего контроль остатков будет работать неверно.

Поэтому необходимо заблокировать остатки от чтения другими транзакциями еще до выполнения второго запроса. Т.е. прежде чем читать что-то, что мы собираемся изменять, нужно запретить чтение этих данных другими транзакциями до окончания введения изменений нами.

## В режиме Конфигуратор

Как это сделать? Давайте посмотрим на свойство нашей конфигурации **Режим управления блокировкой данных**. Он установлен в значение **Управляемый**.

The image shows a 'Свойства' (Properties) dialog box with various configuration options. The 'Совместимость' (Compatibility) section is expanded, and the 'Режим управления блокировкой данных' (Data locking management mode) is set to 'Управляемый' (Managed). The other options in this section are 'Режим автонумерации объектов' (Object auto-numbering mode) set to 'Не освобождать автоматически' (Do not release automatically) and 'Режим совместимости' (Compatibility mode) set to 'Не использовать' (Do not use).

|                                     |                              |
|-------------------------------------|------------------------------|
| Основной язык                       | Русский                      |
| Краткая информация                  |                              |
| Подробная информация                |                              |
| Логотип                             | Открыть                      |
| Заставка                            | Открыть                      |
| Авторские права                     |                              |
| Адрес информации о поставщике       |                              |
| Адрес информации о конфигурации     |                              |
| ▼ Разработка:                       |                              |
| Поставщик                           |                              |
| Версия                              |                              |
| Адрес каталога обновлений           |                              |
| ▼ Справочная информация:            |                              |
| Включать в содержание справки       | <input type="checkbox"/>     |
| Справочная информация               | Открыть                      |
| ▼ Совместимость:                    |                              |
| Режим управления блокировкой данных | Управляемый                  |
| Режим автонумерации объектов        | Не освобождать автоматически |
| Режим совместимости                 | Не использовать              |

Это значит, что нам нужно использовать управляемые блокировки, которые устанавливаются средствами встроенного языка.

Необходимо заблокировать те данные, которые мы собираемся читать и впоследствии изменять. Для этого у наборов записей регистров есть

свойство **БлокироватьДляИзменения**. Вставим этот код перед записью пустых наборов записей.

```
ПО НоменклатураДокумента.Номенклатура = ОстаткиМатериаловОстатки.Материал";  
  
//Установим необходимость блокировки данных  
в регистрах СтоимостьМатериалов и ОстаткиМатериалов.  
  
Движения.СтоимостьМатериалов.БлокироватьДляИзменения = Истина;  
Движения.ОстаткиМатериалов.БлокироватьДляИзменения = Истина;  
  
//Запишем пустые наборы записей, чтобы читать  
остатки без учета данных в документе.  
Движения.СтоимостьМатериалов.Записать();  
Движения.ОстаткиМатериалов.Записать();  
  
Результат = Запрос2.Выполнить();
```

Управляемая блокировка будет установлена в момент записи этих наборов записей, т.е. как раз перед выполнением второго запроса. Что и требовалось.

В режиме 1С: Предприятие

Запустите режим отладки и проверьте работу нового обработчика события **ОбработкаПроведения**, перепроведя все документы Оказание услуги.

### Контрольные вопросы

- ✓ Почему для доступа к массивам данных информационной базы предпочтительнее использовать запросы.
- ✓ Чем отличается оперативное проведение документов от неоперативного.
- ✓ Почему при неоперативном проведении документов не нужно контролировать остатки.
- ✓ Что такое временные таблицы и зачем их использовать.
- ✓ Что такое менеджер запросов.
- ✓ Как программно блокировать данные.