

Лабораторная работа 13. Оптимизация проведения документа

Оказание услуги

Приступим к одному из самых важных разделов – к оптимизации документа **ОказаниеУслуги** и к полному изменению его обработчика события **ОбработкаПроведения**.

Изменения, которые следует внести в документ **ОказаниеУслуги**, будут преследовать три цели:

- повышение скорости выполнения процедуры;
- автоматическое определение стоимости расходуемых материалов при проведении документа;
- разделение алгоритма проведения документа на оперативный и неоперативный режимы и контроль остатков в случае оперативного проведения документа.

Повышение скорости проведения

Первое, чем мы займемся, – избавимся от **вредной** конструкции **ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры**.

В режиме Конфигуратор

Откроем модуль документа **ОказаниеУслуги**. Процедура проведения этого документа выглядит так.

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Движения.Продажи.Записывать = Истина;

Для **Каждого** **ТекСтрокаПереченьНоменклатуры** Из **ПереченьНоменклатуры**
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;
        ДвижениеКоличество = 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.Стоимость;

КонецЕсли;

// РегистрПродажи
        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 все данные, необходимые для проведения документа, мы получаем из самого документа, и только для определения того, чем является номенклатура (товаром или услугой), мы обращаемся к базе данных, читая данные всего объекта **Номенклатура**.

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры

Это не единственные данные, которые **не содержатся** в самом документе и которые в то же время будут нужны нам для правильного его проведения.

Поэтому поступим следующим образом – все данные, связанные с номенклатурой, которая содержится в табличной части документа, мы будем

получать с помощью запроса к базе данных. А данные, связанные с самим документом, например, дата документа или склад, по-прежнему будем получать из документа. Такой подход позволит нам читать только нужные данные и за счет этого максимально ускорить проведение документа.

Итак, с помощью запроса будем получать:

- ✓ номенклатуру,
- ✓ количество,
- ✓ сумму,
- ✓ стоимость.

Из документа возьмем следующие данные:

- ✓ дата,
- ✓ клиент,
- ✓ мастер,
- ✓ склад.

Приступим к созданию запроса. Установим курсор перед циклом обхода табличной части документа и из контекстного меню выберем пункт **Конструктор запроса с обработкой результата**. Подтвердим, что мы хотим создать новый запрос.

В окне конструктора запросов перейдем на закладку **Таблицы и поля** и выберем таблицу **ОказаниеУслугиПереченьНоменклатуры** – это табличная часть документа **ОказаниеУслуги**. Из этой таблицы нам нужны поля – **Номенклатура**, **Номенклатура.ВидНоменклатуры**, **Количество**, **Сумма** и **Стоимость** (рис.13.1).

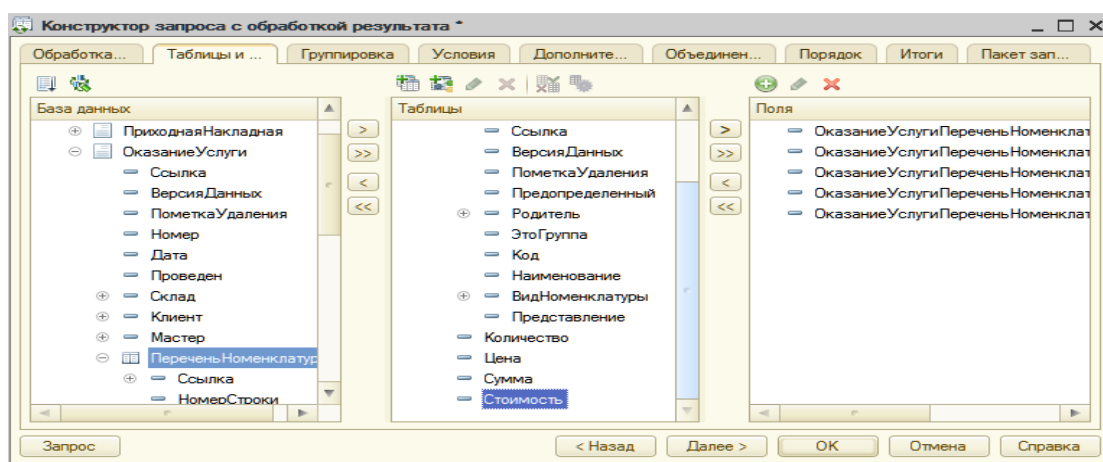


Рис.13.1. Выбранные поля

Но нам нужны не все записи этой таблицы, а только те, которые относятся к

документу **ОказаниеУслуги**. Поэтому перейдем на закладку **Условия** и зададим условие отбора из таблицы документа только строк проводимого документа.

Для этого перетащим поле **Ссылка** в список условий запроса.

ОказаниеУслугиПереченьНоменклатуры.Ссылка = Ссылка

Ссылка на этот документ будет передана в параметр запроса **Ссылка** (рис.13.2).

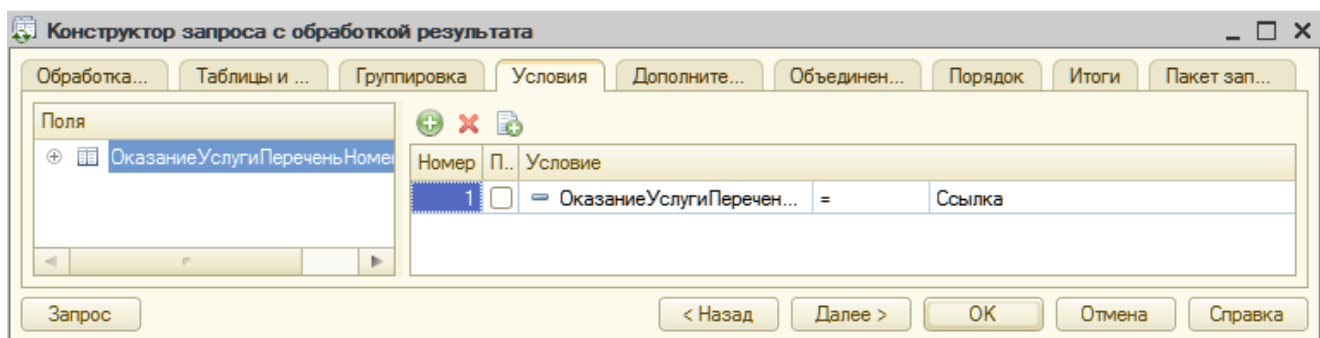


Рис.13.2. Условие отбора из таблицы документа

Также следует учесть, что в табличной части документа одна и та же номенклатура может встречаться несколько раз.

Поэтому на закладке **Группировка** сгруппируем наши записи по полю **Номенклатура** и **НоменклатураВидНоменклатуры**, а рассчитывать будем сумму значений для полей **Количество** и **Сумма**.

Благодаря этому в результате значения номенклатуры повторяться не будут, и для каждого из них будут посчитаны суммарные значения по полям **Количество** и **Сумма**, если в табличной части документа содержится несколько строк с одинаковой номенклатурой.

Также в состав суммируемых полей включим и поле **Стоимость**. По нему будем рассчитывать, например, функцию **Максимум**.

Подразумевается, что для разных строк одной и той же номенклатуры стоимость будет одинаковой, поэтому функция **Максимум** нужна лишь для того, чтобы получить одно из имеющихся значений стоимости (рис.13.3).

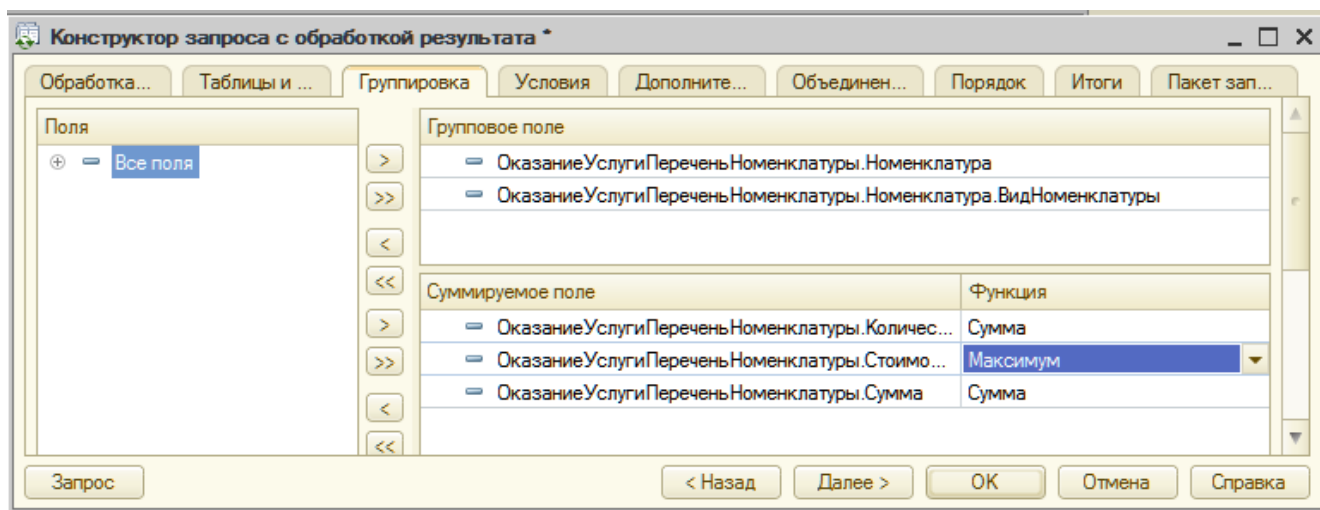


Рис.13.3. Группировка строк таблицы документа

На закладке **Объединения/Псевдонимы** зададим псевдонимы для полей **Количество** и **Сумма** – **КоличествоВДокументе** и **СуммаВДокументе**, а для поля **НоменклатураВидНоменклатуры** – **ВидНоменклатуры** просто для облегчения чтения запроса (рис.13.4).

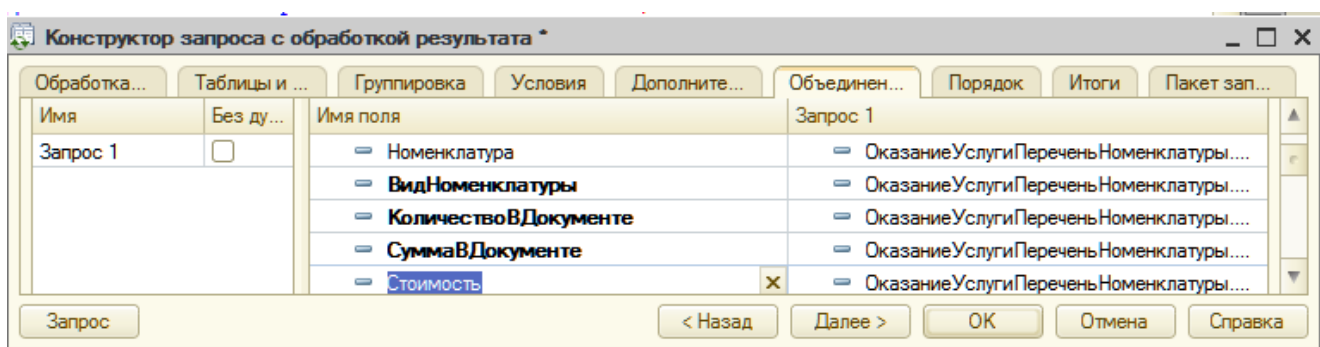


Рис.13.4. Псевдонимы полей

Нажмем **ОК** и посмотрим, какой текст запроса сформировал конструктор.

```
//{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,
|
    МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость)    КАК
Стоимость,
|
    СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество)    КАК
КоличествоВДокументе,
|СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК
СуммаВДокументе
|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
КонецЦикла;
//}} КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

```

Конструктор написал код для выполнения и обхода записей запроса. Рассмотрим этот код.

Для работы с запросами используется, как известно, 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 запрос выполняется – **Запрос.Выполнить()**, получается объект **РезультатЗапроса**, и выполняется его метод **Выбрать()**, который формирует выборку записей из результата запроса.

Таким образом, получается объект **ВыборкаИзРезультатаЗапроса**, который помещается в переменную **ВыборкаДетальныеЗаписи**.

Далее, используя метод **Следующий()** этого объекта – **ВыборкаДетальныеЗаписи.Следующий()**, мы в цикле обходим выборку записей запроса. Выполняя метод выборки запроса, указатель на каждом шаге цикла позиционируется 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 ранее в этом модуле строки, описывающие движения регистров, внутрь цикла обхода результата запроса.

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

// Вставить обработку выборки ВыборкаДетальныеЗаписи

КонецЦикла;

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

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.Стоимость
            * ТекСтрокаПереченьНоменклатуры.Количество;
КонецЕсли;

```

В условии заменим **ТекСтрокаПереченьНоменклатуры.Номенклатура** на **ВыборкаДетальныеЗаписи**, так как вид номенклатуры мы теперь получаем из запроса. В движениях также заменим **ТекСтрокаПереченьНоменклатуры** на **ВыборкаДетальныеЗаписи**.

```

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.Количество;

```


//регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
ДвижениеСтоимость = ВыборкаДетальныеЗаписи.Стоимость
* ВыборкаДетальныеЗаписи.Количество;

КонецЕсли;

КонецЦикла;

Так как для поля **Количество** мы задали псевдоним в запросе, поэтому заменим его на **КоличествоВДокументе**.

Пока ВыборкаДетальныеЗаписи.СледующийО 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;
ДвижениеСклад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.**КоличествоВДокументе**;

// регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
ДвижениеСтоимость = ВыборкаДетальныеЗаписи.**КоличествоВДокументе**
* ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

КонецЦикла; I

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

Пока ВыборкаДетальныеЗаписиСледующий() **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;
ДвижениеСклад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписиКоличествоВДокументе;

// 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;

// РегистрПродажи

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**.

Пока ВыборкаДетальныеЗаписиСледующий() **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;

ДвижениеСклад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписиКоличествоВДокументе;

// 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;

// РегистрПродажи

Движение = Движения.Продажи.Добавить();

Движение.Период = Дата;

Движение.Номенклатура = **ВыборкаДетальныеЗаписи.Номенклатура;**

Движение.Клиент = Клиент;

Движение.Мастер = Мастер;

Движение.Количество = **ВыборкаДетальныеЗаписи.Количество;**

Движение.Выручка = **ВыборкаДетальныеЗаписи.Сумма;**

Движение.Стоимость = **ВыборкаДетальныеЗаписи.Стоимость**
*** ВыборкаДетальныеЗаписи.Количество;**

КонецЦикла;

В движениях по регистру продажи поля запроса **Сумма** и **Количество**
заменим на **СуммаВДокументе** и **КоличествоВДокументе**.

| **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;

ДвижениеСклад = 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;

// РегистрПродажи

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

В режиме 1С:Предприятие

Теперь нужно запустить 1С:Предприятие в режиме отладки, перепровести документы Оказание услуги и проверить, что ничего не изменилось.

Таким образом, мы выполнили **первый пункт плана** – избавились в процедуре проведения от считывания всех данных объекта **Номенклатура** и, тем самым, оптимизировали выполнение процедуры проведения.

Автоматический расчет стоимости

Теперь приступим ко **второму этапу** плана.

До сих пор стоимость расходуемых 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 то количество этой номенклатуры, которое у нас имеется, и таким образом получать среднюю стоимость одной единицы этой номенклатуры.

Чтобы выполнить такой расчет, нам нужны дополнительные данные, которых у нас сейчас нет.

Для каждой номенклатуры из табличной части нам понадобятся:

- ✓ стоимость, хранящаяся в регистре **СтоимостьМатериалов**;
- ✓ общее количество на всех складах, хранящееся в регистре **ОстаткиМатериалов**.

Поэтому нам нужно будет доработать запрос таким образом, чтобы он получал из базы данных также и эти данные.

Таким образом, нам нужно, чтобы запрос возвращал для каждой номенклатуры документа следующие поля:

- номенклатура,
- количество в документе,
- сумма в документе,
- вид номенклатуры,
- стоимость,
- количество на всех складах.

Первые четыре поля мы получаем из табличной части самого документа, а вот **последние два** нужно будет получить из других таблиц базы данных:

- стоимость – из регистра **СтоимостьМатериалов**;
- остатки на всех складах – из регистра **ОстаткиМатериалов**.

Это значит, что 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 номенклатуры абсолютно для **всей номенклатуры**. А нас интересует только та номенклатура, которая **указана в документе**. Эта деталь очень важна для реальной большой базы. Например, в справочнике **Номенклатура** – 15000 наименований, а в документе – всего 5 наименований. Получится, что виртуальная таблица остатков будет рассчитывать стоимость или остатки по всем 15000 наименованиям номенклатуры. А в результате, когда мы левым соединением станем соединять ее с таблицей документа, из этих 15000 строк стоимости или остатков мы возьмем всего лишь 5 строк – для той номенклатуры, которая указана в документе. Остальные 14 995 строк будут просто отброшены – система напрасно их рассчитывала.

Естественно, такой запрос является очень неоптимальным. Поэтому во все виртуальные таблицы, которые мы будем использовать, нужно **добавить условие отбора только той номенклатуры, которая содержится в табличной части документа**. В этом случае стоимость и остатки будут рассчитаны не для всей номенклатуры вообще, а только для нужной номенклатуры.

Кроме этого, из таблиц документа тоже берутся не все данные, а только данные, относящиеся к одному конкретному документу. Чтобы не получать этот список три раза – для документа и каждой виртуальной таблицы, – мы сформируем его **заранее** и затем будем использовать в нужных условиях запроса.

Выполнить эту задачу нам помогут **временные таблицы**.

Временные таблицы – это программные объекты, которые разработчик создает и заполняет данными, а запросы используют эти данные временных таблиц для своих нужд. Например, для наложения некоторого сложного условия, как в нашем случае.

Таким образом, схема **Запроса** приобретает следующий вид (рис. 13.5).



Рис.13.5. Схема запроса

В режиме Конфигуратор

Сначала удалим реквизит табличной части **Стоимость** документа **ОказаниеУслуги**, который нам больше не понадобится. Для этого откроем в конфигураторе окно редактирования объекта конфигурации **Документ ОказаниеУслуги**, перейдем на закладку **Данные**, раскроем список реквизитов табличной части документа, выделим реквизит **Стоимость** и нажмем кнопку **Удалить** в командной панели.

Также следует удалить соответствующее поле из таблицы **ПереченьНоменклатуры**, расположенной в форме. Для этого откроем форму **ФормаДокумента** документа **ОказаниеУслуги** и в окне структуры элементов формы выделим поле таблицы **ПереченьНоменклатурыСтоимость** и нажмем кнопку **Удалить** в командной панели (рис.13.6).

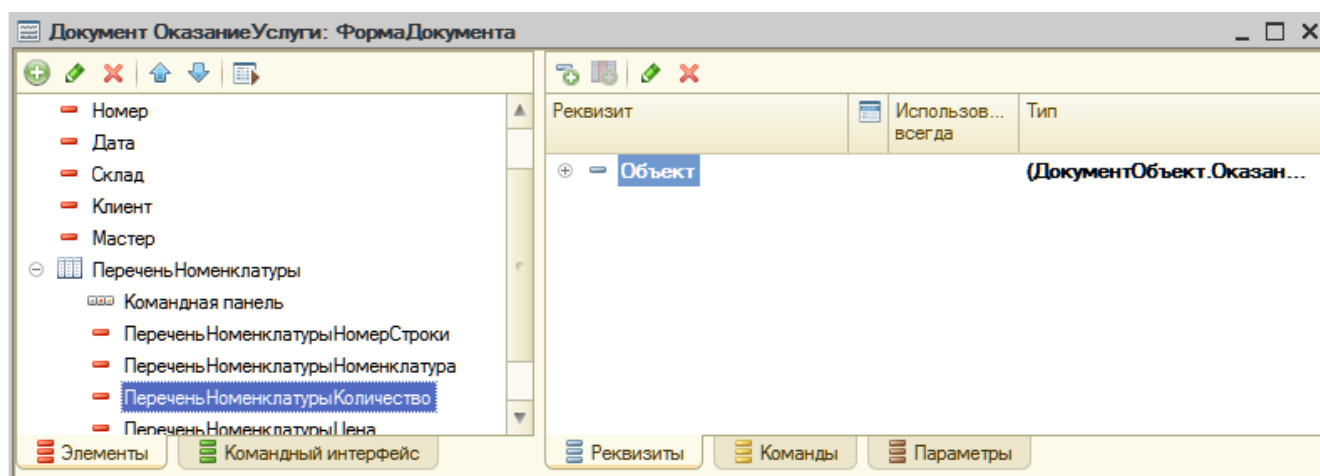


Рис.13.6. Удаление поля табличной части

Теперь перейдем к запросу. Временную таблицу сформируем с помощью того запроса, который уже написан.

Откроем модуль документа **ОказаниеУслуги**. В процедуре **ОбработкаПроведения()** перед созданием запроса создадим менеджер временных таблиц и укажем, что запрос будет использовать этот созданный менеджер временных таблиц.

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

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

Чтобы конструктор запроса смог открыть наш запрос, удалим из него строку, так как поля **Стоимость** больше нет.

|МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость)
КАК Стоимость

Также удалим запятую в конце предыдущей строки.

|СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма)
КАК СуммаВДокументе

Теперь установим курсор внутри текста запроса, например, на слове **ВЫБРАТЬ**, и выполним команду контекстного меню **Конструктор запроса**. Существующий текст запроса будет показан в форме конструктора запросов (рис.13.7).

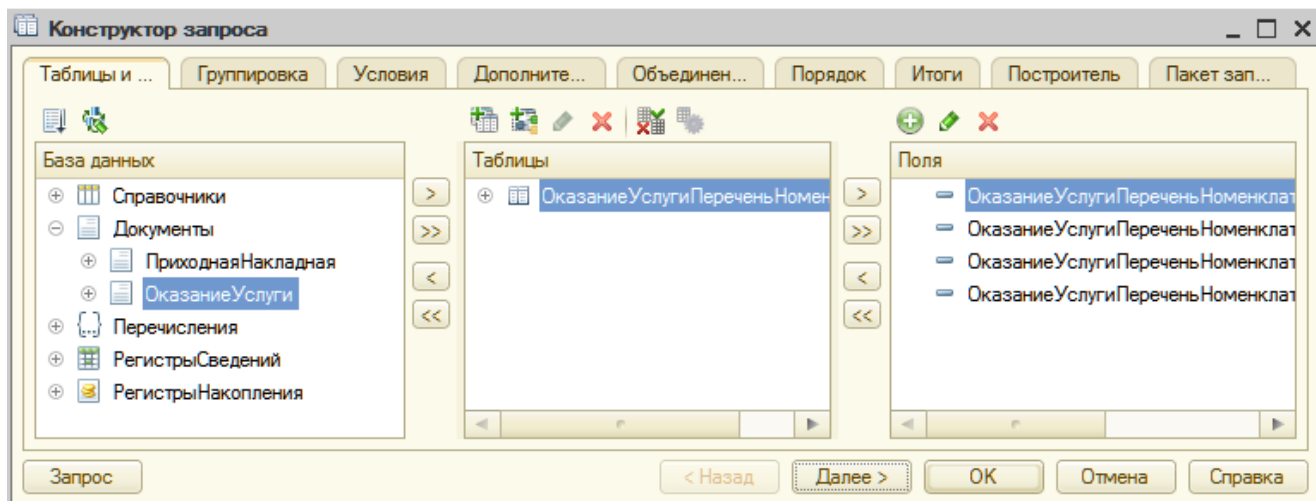


Рис.13.7. Конструктор запроса

Чтобы результат запроса поместить во временную таблицу, перейдем на закладку **Дополнительно** и нажмем радиокнопку **Создание временной таблицы**.

Зададим имя временной таблицы – **НоменклатураДокумента** (рис. 13.8).

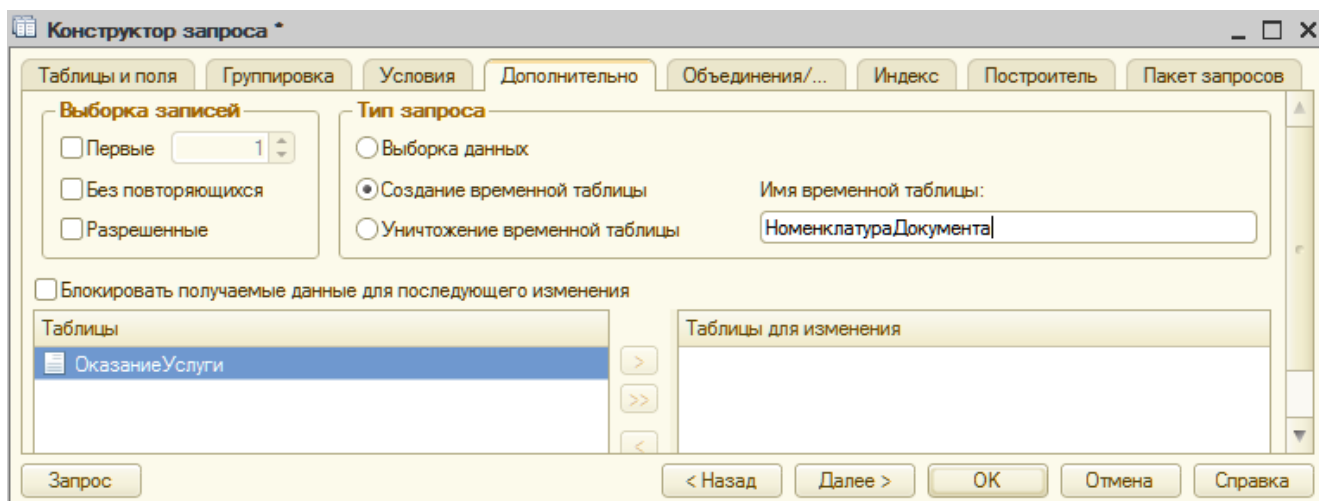


Рис.13.8. Создание временной таблицы

"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";

Строка **ПОМЕСТИТЬ НоменклатураДокумента** означает, что результат запроса будет сохранен во временной таблице **НоменклатураДокумента**. Таким

образом, мы выполнили первую часть плана – создали запрос, помещающий данные табличной части документа во временную таблицу (рис.13.5).

Теперь если для второго запроса укажем этот же самый менеджер временных таблиц **МенеджерВТ**, то в этом запросе мы сможем обратиться к данным той же временной таблицы.

Займемся конструированием второго запроса. Установим курсор на следующую строку после оператора **Результат = Запрос.Выполнить()**; Именно здесь выполняется создание временной таблицы. Напишем заготовку будущего запроса.

Запрос2 = Новый Запрос;

Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;

Запрос2.Текст = "";

Мы создали новый объект **Запрос2** и назначили ему тот же самый менеджер временных таблиц, чтобы иметь возможность обращаться к созданной ранее временной таблице.

Теперь установим курсор внутрь кавычек, выполним команду контекстного меню **Конструктор запроса** и согласимся на создание нового запроса.

Поскольку мы собираемся выбирать данные из временной таблицы, создадим в запросе описание этой временной таблицы. Для этого над списком **Таблицы** нажмем кнопку **Создать описание временной таблицы** (рис. 13.9).

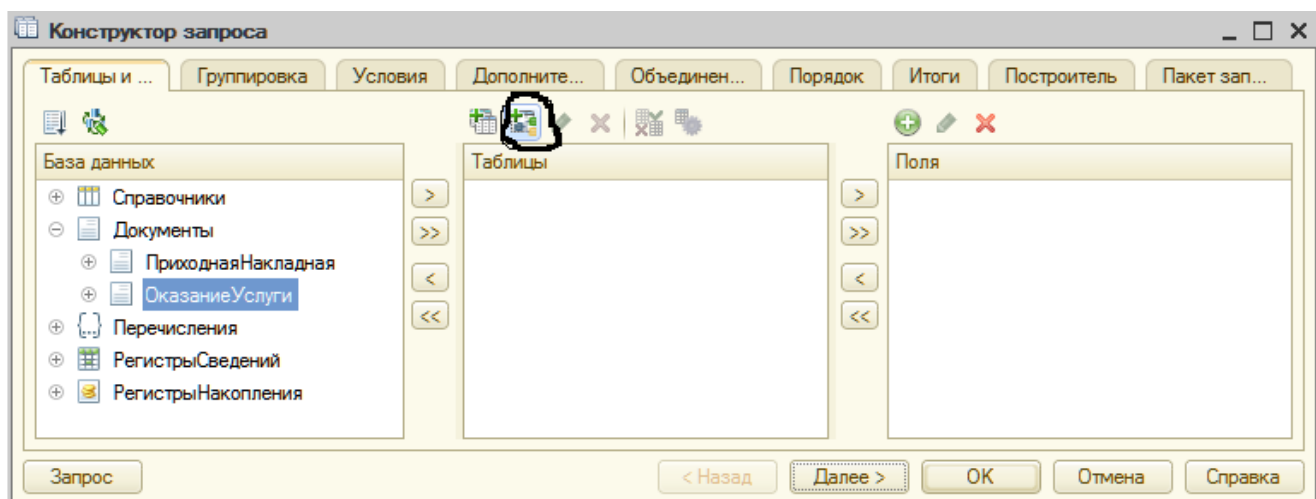


Рис. 13.9. Создание описания временной таблицы

В открывшемся окне введем имя временной таблицы – **НоменклатураДокумента** и добавим описание полей:

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

Нажмем ОК. Получилось описание временной таблицы (рис. 13.10).

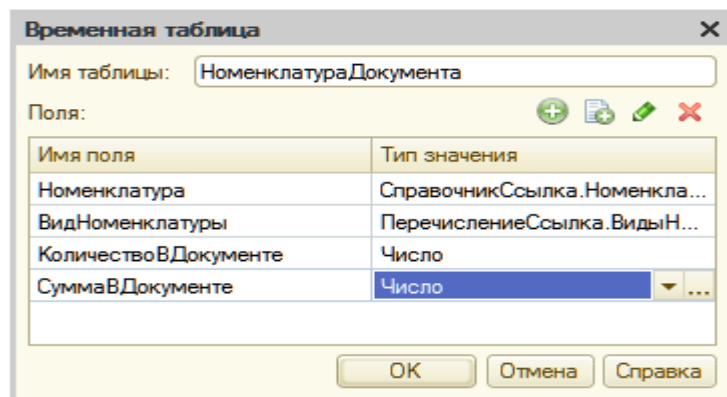


Рис. 13.10. Создание описания временной таблицы

Выберем из этой таблицы все поля и нажмем кнопку **Запрос**. Текст запроса:

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 (рис. 13.5). Теперь соединим эту конструкцию левыми соединениями с таблицами остатков.

Начнем со стоимости материалов.

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, а поле **Номенклатура** временной таблицы должно быть равно полю **Материал** таблицы остатков (рис. 13.11).

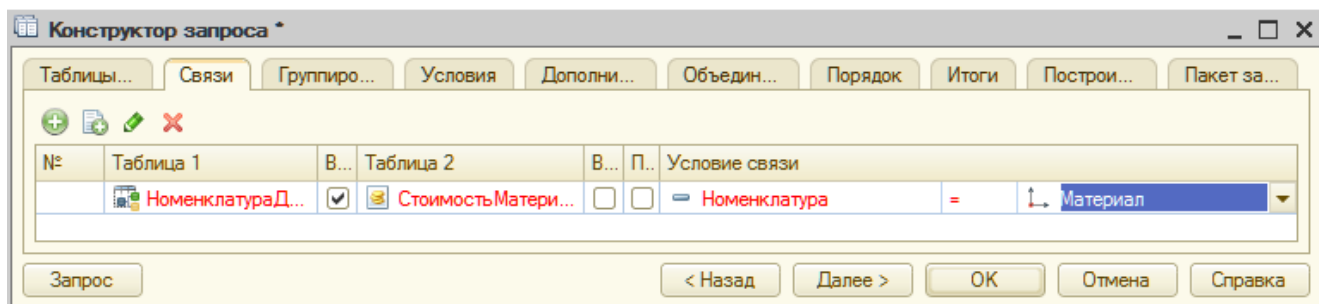


Рис. 13.11. Связи между таблицами

Теперь нужно ограничить виртуальную таблицу только той номенклатурой, которая есть во временной таблице. Поэтому вернемся на закладку **Таблицы и поля**, выделим в списке таблицу **СтоимостьМатериаловОстатки** и нажмем кнопку **Параметры виртуальной таблицы**, расположенную над списком таблиц.

Зададим параметр **Условие** следующим образом.

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента). То есть материал должен быть среди номенклатуры, выбранной из временной таблицы.

Нажмем кнопку **Запрос** и посмотрим текст запроса, который сформировал конструктор.

ВЫБРАТЬ

НоменклатураДокумента.Номенклатура,

НоменклатураДокумента.ВидНоменклатуры,

НоменклатураДокумента.КоличествоВДокументе,

НоменклатураДокумента.СуммаВДокументе

ИЗ

НоменклатураДокумента КАК НоменклатураДокумента

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.

Остатки(, Материал В

(ВЫБРАТЬ

НоменклатураДокумента.Номенклатура

ИЗ

НоменклатураДокумента))

КАК СтоимостьМатериаловОстатки

ПО НоменклатураДокумента.Номенклатура=

СтоимостьМатериаловОстатки.Материал

То есть мы добавили к выбранным ранее полям стоимость номенклатуры (рис. 13.5).

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

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 (рис. 13.12).

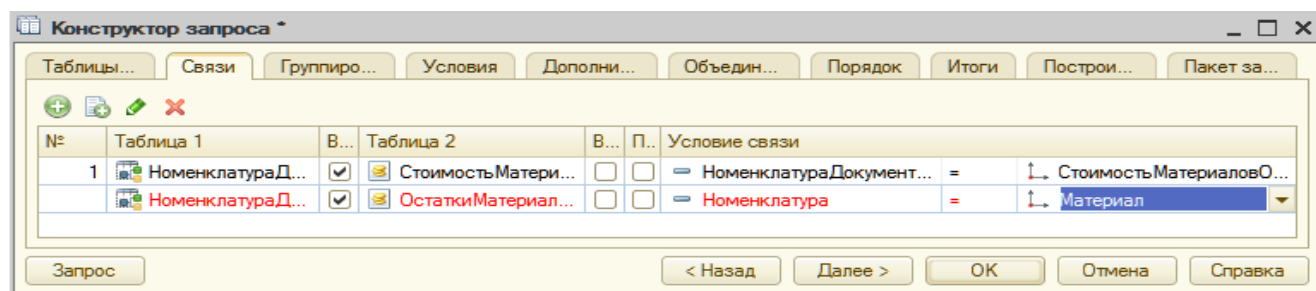


Рис. 13.12. Связи между таблицами

Также зададим параметры виртуальной таблицы **ОстаткиМатериаловОстатки**. В параметр **Условие** внесем следующий текст **Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)**

Получим следующий текст запроса.

ВЫБРАТЬ

НоменклатураДокумента.Номенклатура,
 НоменклатураДокумента.ВидНоменклатуры,
 НоменклатураДокумента.КоличествоВДокументе,
 НоменклатураДокумента.СуммаВДокументе,
 СтоимостьМатериаловОстатки.СтоимостьОстаток,
 ОстаткиМатериаловОстатки.КоличествоОстаток

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

То есть мы добавили к выбранным ранее полям остатки номенклатуры на всех складах (рис. 13.5).

В заключение перейдем на закладку **Объединения/Псевдонимы** и зададим следующие псевдонимы полей (рис. 13.13):

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

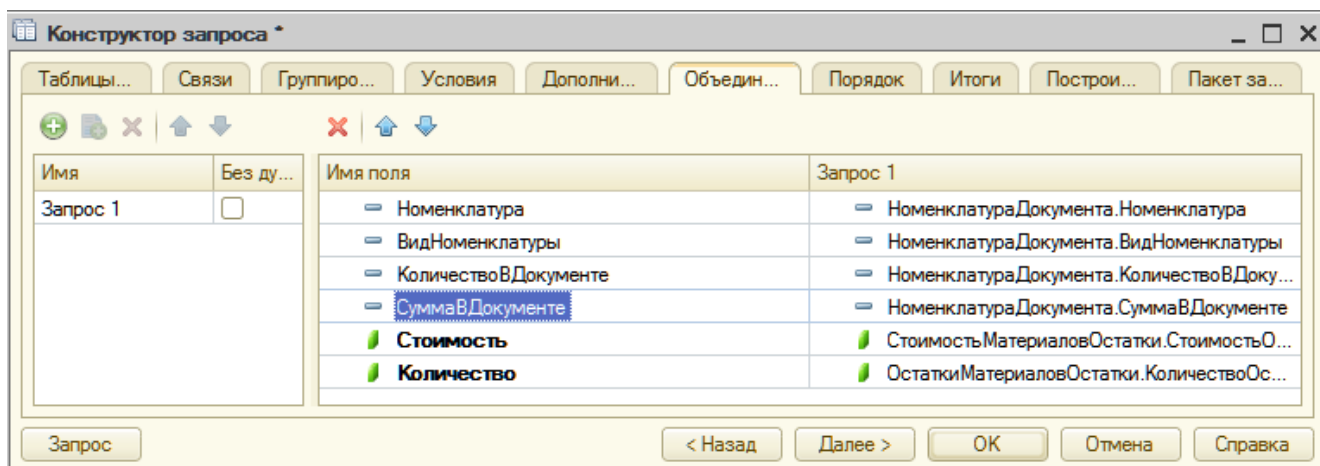


Рис. 13.13. Псевдонимы полей

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

При работе с запросом нужно предусмотреть ситуацию, когда номенклатура в справочнике есть, но у нее нет ни остатков, ни стоимости. Это может быть в том случае, когда номенклатуру создали в справочнике, но она еще не поступала в фирму. При этом левые соединения с виртуальными таблицами не вернут ничего. На языке запросов это значит, что в полях **Стоимость** и **Количество** будут значения **NULL**.

Поэтому в запросе избавимся от этих значений. Для этого 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.

Перейдем на закладку **Таблицы и поля**, выделим поле **СтоимостьМатериаловОстатки.СтоимостьОстаток** и нажмем кнопку **Изменить текущий элемент** (рис. 13.14).

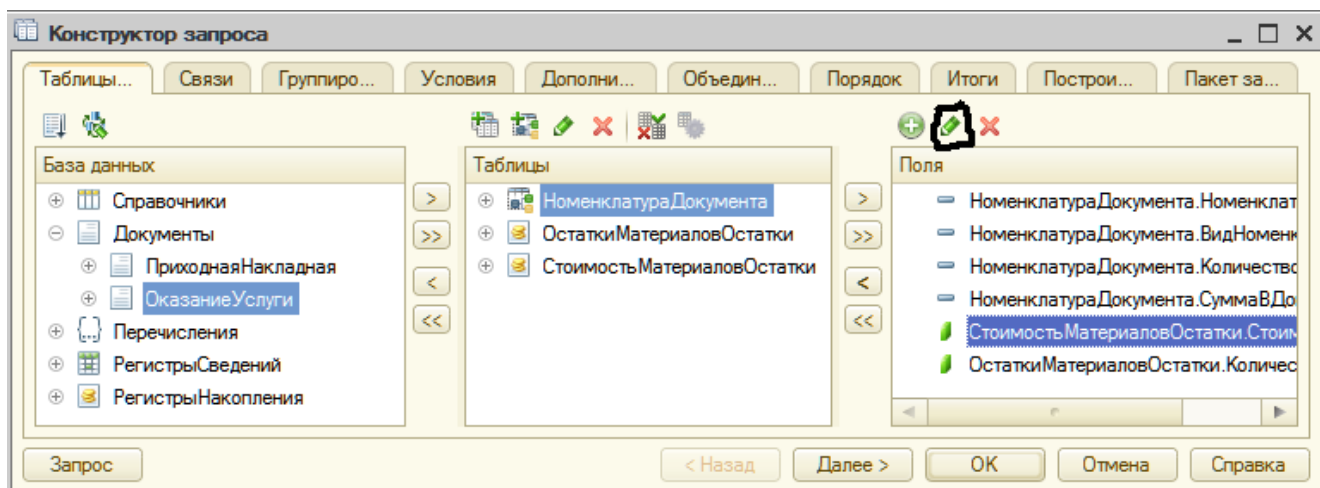


Рис. 13.14. Изменение значения поля в запросе

В открывшемся окне (рис. 13.15) отредактируем значение поля следующим образом **ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0)** и нажмем **ОК**.

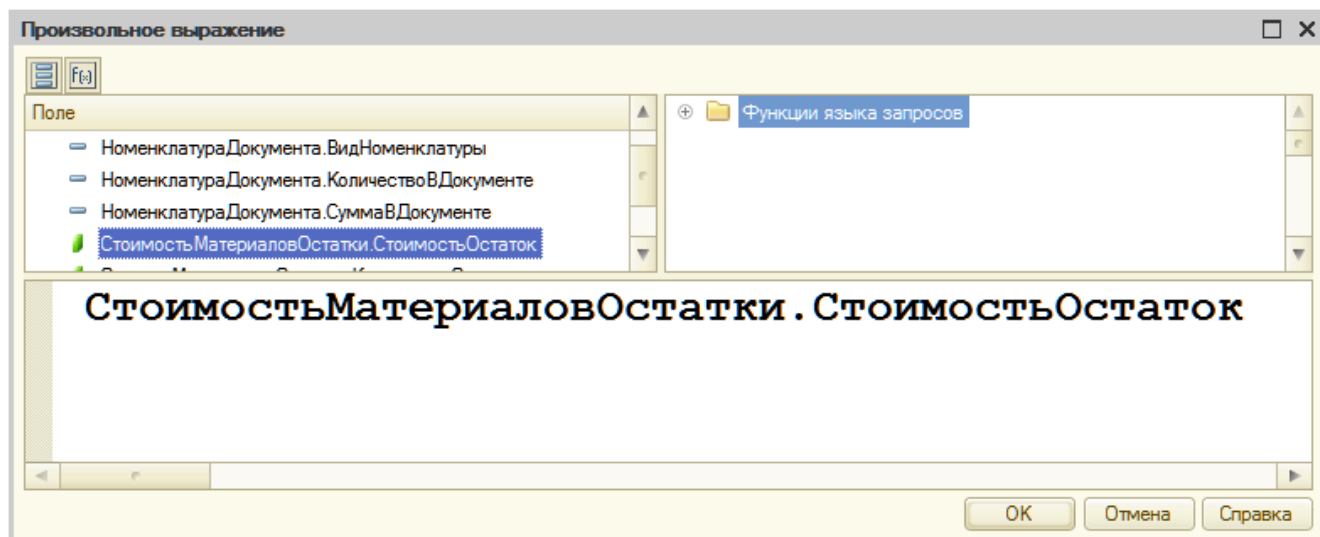


Рис. 13.15. Изменение значения поля в запросе

Аналогично поступим и с другим полем:
ОстаткиМатериаловОстатки.КоличествоОсгatok.

Нажмем **ОК**. Текст запроса будет вставлен в модуль.

Запрос2. Текст =

"ВЫБРАТЬ

|НоменклатураДокумента.Номенклатура,

|НоменклатураДокумента.ВидНоменклатуры,

|НоменклатураДокумента.КоличествоВДокументе,

|НоменклатураДокумента.СуммаВДокументе,

|ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0) КАК

Стоимость,

|ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток, 0) 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(
| ,
|Mатериал В
|(ВЫБРАТЬ
| НоменклатураДокумента.Номенклатура
|ИЗ
|НоменклатураДокумента)) КАК ОстаткиМатериаловОстатки
|ПО НоменклатураДокумента.Номенклатура =
                        ОстаткиМатериаловОстатки.Материал";

```

Остается лишь дописать после запроса оператор его выполнения.

```
Запрос2 = Новый Запрос;
```

```
Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;
```

```
Запрос2.Текст = "ВЫБРАТЬ
```

```
| НоменклатураДокумента.Номенклатура,
```

```
.....
```

```
| ПО НоменклатураДокумента.Номенклатура =
```

```
                        СтоимостьМатериаловОстатки.Материал";
```

```
Результат = Запрос2.Выполнить();
```

```
ВыборкаДетальныеЗаписи = Результат.Выбрать();
```

Теперь разберемся с записью движений. Все операторы, которые были написаны ранее, будут работать без изменений. Единственное, что потребуется изменить, – это способ получения стоимости. Раньше мы просто брали ее из документа, теперь же нужно ее рассчитать на основании тех данных, которые мы получили с помощью запроса.

Стоимость материала равна частному от деления всей стоимости, полученной с помощью запроса – **Стоимость** – на общее количество материала на всех складах – **Количество**.

Но, как мы уже установили, возможна ситуация, когда поле **Количество** будет равно 0, а на 0 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.Количество;

//регистр СтоимостьМатериалов Расход

совершены в будущем. Поэтому задача неоперативного проведения документов – просто отразить в информационной базе данные о совершенных операциях.

При неоперативном проведении документов не имеет смысла производить целый ряд проверок, в частности контроль остатков. Подразумевается, что если в процессе неоперативного проведения документов были допущены ошибки, например, списано такое количество номенклатуры, которого не было на складе на дату проведения документа, то анализ полученного состояния базы данных является отдельной задачей, не относящейся к неоперативному проведению.

Таким образом, **оперативное проведение** служит для того, чтобы в реальном режиме многопользовательской работы определить возможность или невозможность выполнения той или иной операции. **Неоперативное проведение** предназначено для безусловного отражения в базе операций, которые уже были совершены или точно будут совершены.

С оперативным проведением документов связано понятие **оперативной отметки времени** и понятие **момента времени**.

Контроль остатков

Общая методика контроля остатков при проведении документа заключается в следующем: сначала нужно записать движения документа, а затем, когда движения уже записаны, прочитать из базы данных остатки.

Если появились отрицательные остатки, значит, такой документ проводить нельзя. Нужно сообщить пользователю, каких материалов не хватает, и отменить проведение документа.

Если же отрицательных остатков не появилось, тогда можно смело проводить документ.

Полдела у нас уже сделано: мы формируем движения документа и записываем их. Единственное, что нам осталось, в случае оперативного проведения проконтролировать, **что получилось**, и если появились отрицательные остатки, отменить проведение документа.

В режиме Конфигуратор

Сделаем заготовку. После цикла обхода результата запроса и перед концом процедуры напишем следующие строки.

.....

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.

Так как нам придется снова получать остатки только для той номенклатуры, которая в документе, укажем, что этот запрос будет использовать тот же самый менеджер временных таблиц **МенеджерВТ**.

Если Режим = РежимПроведенияДокумента.Оперативный Тогда
 // Проверить отрицательные остатки
ЗапросЗ = Новый Запрос;
ЗапросЗ.МенеджерВременныхТаблиц = МенеджерВТ;
ЗапросЗ.Текст = "";
КонецЕсли;

Установим курсор внутрь кавычек и вызовем конструктор запроса. Подтвердим создание нового запроса.

Выберем таблицу **ОстаткиМатериалов.Остатки** и из нее два поля: **Материал** и **КоличествоОстаток**.

Зададим параметры этой таблицы. Для этого выделим в списке таблиц таблицу **ОстаткиМатериалов.Остатки** и нажмем над списком таблиц кнопку **Параметры виртуальной таблицы**. В параметре **Условие** напишем:

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

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

Затем на закладке **Условия** перенесем в список условий поле **ОстаткиМатериаловОстатки.КоличествоОстаток**, поставим флажок **Произвольное** и укажем, что нас интересуют только отрицательные остатки.

ОстаткиМатериаловОстатки.КоличествоОстаток < 0

Нажмем **ОК**. Текст запроса будет выглядеть следующим образом.

// 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";

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

Запрос3 = Новый Запрос;

Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;

Запрос3.Текст = "ВЫБРАТЬ

|

|ОстаткиМатериаловОстатки.Материал,

|
|

|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;

Поясним добавленный текст.

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** процедуры проведения документа присваивается значение **Истина**, то есть документ не проводится. Начатая транзакция отменяется, и данные, измененные в процессе проведения, возвращаются в исходное состояние, существовавшее до начала проведения документа.

Блокировка данных, которые читаются и изменяются при проведении

Имеется еще один важный момент, который мы пропустили.

Сейчас схема процедуры обработки проведения такова:

1. Выполняется первый запрос с именем **Запрос**. Результат – формируется временная таблица из перечня номенклатуры документа.
2. Выполняется второй запрос с именем **Запрос2**. Результат – читается стоимость и остатки для номенклатуры, содержащейся в табличной части документа.
3. Записываются движения регистров – **Движения.Записать()**.
4. Выполняется третий запрос с именем **Запрос3**. Результат – проверяется наличие отрицательных остатков.

Начиная с выполнения второго запроса и до конца процедуры, необходимо обеспечить неизменность стоимости и остатков номенклатуры, с которой мы работаем, и запретить другим транзакциям даже читать эти данные. Сама система, естественно, заблокирует изменение этих данных, но лишь начиная с того момента, когда мы запишем движения.

Однако может возникнуть следующая ситуация. Выполняя второй запрос, мы прочитали, что есть 2 шт. некоторого материала. Другой пользователь, который собирается списывать материалы, тоже прочитал, что есть 2 шт. этого материала. После этого мы записали движения, и система заблокировала эти данные. Транзакция, которую инициировал другой пользователь, 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 п. 3 и п. 4, в результате чего контроль остатков будет работать неверно.

Поэтому, чтобы не происходило таких коллизий, нам необходимо заблокировать остатки от чтения другими транзакциями еще до выполнения второго запроса. То есть прежде чем читать что-то, что мы собираемся изменять, нужно запретить чтение этих данных другими транзакциями до тех пор, пока мы не закончим свои изменения или не откажемся от проведения документа.

В режиме Конфигуратор

Посмотрим на свойство нашей учебной конфигурации **Режим управления**

блокировкой данных. Он установлен в значение **Управляемый**. 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**, которым мы и воспользуемся.

Вставим этот код перед записью пустых наборов записей.

.....

```
[ПО НоменклатураДокумента.Номенклатура =  
СтоимостьМатериаловОстатки.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, то есть как раз перед выполнением второго запроса, что и требовалось.

В режиме 1С:Предприятие

Запустим 1С Предприятие в режиме отладки и проверим работу нового обработчика события **ОбработкаПроведения**, перепроведя все документы **Оказание услуги**. С точки зрения пользователя все работает точно так же, но проведение документов организовано методически правильно и более эффективно с точки зрения доступа к данным.

Контрольные вопросы

1. Почему для доступа к массивам данных информационной базы предпочтительнее использовать запросы?
2. Что такое момент времени?
3. Чем отличается оперативное проведение документов от неоперативного?
4. Что такое оперативная отметка времени?
5. Как запросом получить остатки регистра накопления?
6. На что следует обращать внимание при задании параметров виртуальных таблиц запросов?
7. Почему при неоперативном проведении документов не нужно контролировать остатки?
8. Что такое временные таблицы и зачем их использовать?
9. Что такое менеджер запросов?
10. Как и зачем можно использовать временные таблицы в параметрах виртуальных таблиц?
11. Как программно блокировать данные?