**Лабораторная работа №8**

**Тема:** Оборотные регистры накопления, последовательности, нумераторы, регистры сведений

**Цель работы:** Исследовать особенности работы с оборотными регистрами накопления, с регистрами сведений, последовательностями и нумераторами. Рассмотреть методы редактирования управляемых форм.

### Оборотные регистры накопления

Ранее мы работали с регистром накопления, основной задачей которого было – предоставление данных об остатках материалов. Приходные данные в этот регистр попадали при поступлении материалов в организацию, они были "привязаны" к определенному ответственному лицу. Расходные данные в регистр вносились тогда, когда материалы отпускались мастеру для производственных целей.

Организация, работу которой мы автоматизируем, помимо контроля остатков материалов на складе, хотела бы организовать учет продажи материалов и учет оказания услуг с возможностью подсчета прибыли от этих операций. Для нахождения показателя прибыли, например, при продаже материалов, нам достаточно знать себестоимость этих материалов и продажную цену. Для нахождения показателя прибыли по оказанным услугам, мы должны, во-первых, знать стоимость оказанных услуг, во-вторых – знать показатель их себестоимости. Показатели себестоимости обычно включают в себя множество данных – заработную плату сотрудников, амортизацию основных средств, общехозяйственные и общепроизводственные расходы (арендную плату, например). Мы упростим задачу, решив, что себестоимость услуг заключается в себестоимости материалов, израсходованных на оказание этих услуг. Данные по прибыли от продаж нам хотелось бы получать ежедневно, а данные по прибыли от оказания услуг – ежемесячно.

Подобные результирующие показатели в реальной системе учета могут, например, влиять на заработную плату сотрудников. К тому же, анализ таких показателей в разрезе контрагентов, которым проданы материалы или которым оказаны услуги, поможет выделить среди контрагентов тех, кто приносит организации наибольшую прибыль, а, значит, может нуждаться в особом внимании.

Для реализации этих задач нам, во-первых, понадобится документ (или документы), для фиксации в системе продаж материалов и оказания услуг, а, во-вторых – объекты для хранения сведений о проданных материалах, их себестоимости, об оказанных услугах и о материалах, которые были затрачены на оказание этих услуг.

В осуществлении этого замысла нам помогут оборотные регистры накопления. Конечно, показатели оборотов мы можем получать и сейчас, так как регистр с видом Остатки хранит и остатки, и обороты, но, если показатель остатков нам не важен (например, нам нет смысла считать "остатки" по количеству оказанных услуг), предпочтительнее будет использовать регистр с видом **Обороты**. Частота получения данных из регистра так же имеет значение, так как оборотные регистры накопления поддерживают механизм агрегатов, который позволяет ускорить получение данных за определенные периоды, по определенному набору измерений (разрезов) регистра.

Здесь мы ограничимся разработкой документа реализации материалов и созданием оборотного регистра накопления для хранения данных по реализации и последующего построения на его основе отчетов.

Создадим документ **РеализацияМатериалов**. Добавим его в состав подсистемы **ОперативныйУчетМатериалов**. Запретим **оперативное проведение** документа.

В состав данных документа включим следующие реквизиты:

**Имя**: Покупатель: Тип: СправочникСсылка.Контрагенты

**Имя**: ОтветственныйСотрудник: Тип: СправочникСсылка.Сотрудники

Добавим в документ табличную часть **Материалы** со следующими реквизитами:

**Имя**: Номенклатура, Тип: СправочникСсылка.Номенклатура

**Имя**: Остаток, Тип: Число, Длина 10, точность 3

**Имя**: Себестоимость, Тип: Число, длина 10, точность 2

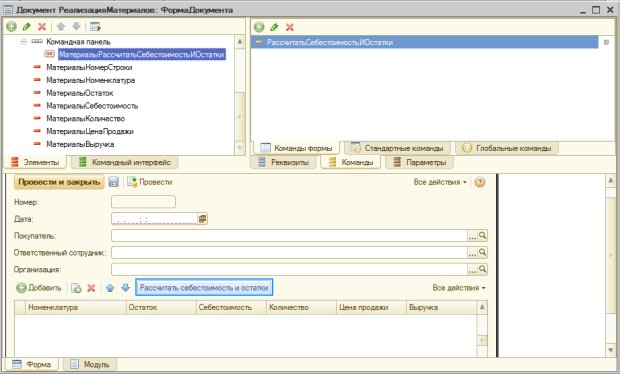
**Имя**: Количество, Тип: Число, длина 10, точность 3

**Имя**: ЦенаПродажи, Тип: Число, длина 10, точность 2

**Имя**: Выручка, тип: Число, длина 10, точность 2

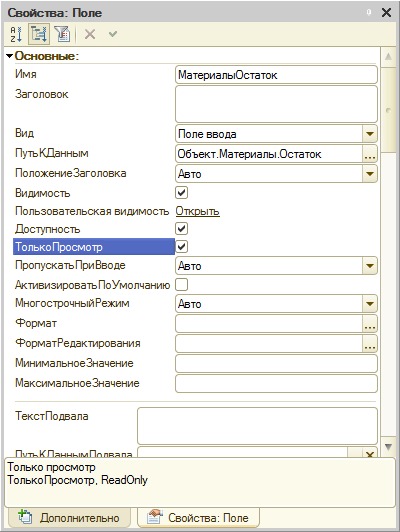
Теперь займемся формой документа. Нам хотелось бы реализовать следующую функциональность – при заполненной данными о номенклатуре табличной части, по нажатию на кнопку **РассчитатьСебестоимостьИОстатки**, заполнять сведения о себестоимости материалов и об их остатках по выбранному ответственному сотруднику, исходя из данных, хранящихся в регистре накопления **ОстаткиМатериалов**.

Создадим форму документа, добавим команду формы **РассчитатьСебестоимостьИОстатки** и переместим ее на командную панель табличного поля, [рис. 8.1](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=1#image.8.1).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_01.jpg)

**Рис. 8.1.** Конструирование формы документа РеализацияМатериалов

В полях **Остаток** и **Себестоимость** пользователь должен увидеть текущие сведения об остатках и себестоимости выбранной номенклатуры, которая числится за выбранным в реквизите документа ответственным лицом. Эти поля должны быть недоступны для редактирования, так как играют лишь вспомогательную, информационную функцию. Для этого у полей **МатериалыОстаток** и **МатериалыСебестоимость** установим свойство **ТолькоПросмотр**, [рис. 8.2](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=1#image.8.2)



**Рис. 8.2.** Запрет редактирования поля пользователем

Теперь займемся кодом, отвечающим за получение данных по остаткам и себестоимости материалов, которые находятся в табличной части. Действия по получению данных и по заполнению ими табличного поля мы должны выполнять на сервере, для этого создадим серверную процедуру **РассчитатьНаСервере()** и вызовем ее из процедуры обработчика **РассчитатьСебестоимостьИОстатки()**.

Для того, чтобы получить нужные данные, нам понадобится запрос, который выбирает данные из табличной части и из регистра накопления **ОстаткиМатериалов**, сгруппировав их по номенклатуре. Воспользуемся уже знакомой вам консолью запросов для того, чтобы построить подобный запрос. Здесь следует вспомнить, что, конструируя процедуру проведения документа **ОтпускМатериаловМастеру**, мы строили подобный запрос. Поэтому сейчас мы можем просто этот запрос доработать. Дорабатывать его, однако, удобно с помощью консоли запросов. Перенесем текст запроса из кода метода **ОбработкаПроведения** документа **ОтпускМатериаловМастеру**, добавим в поле текста запроса консоли запросов в режиме 1С:Предприятие и займемся редактированием текста.

Мы начинаем с такого текста запроса:

ВЫБРАТЬ

ДокМ.Номенклатура,

СУММА(ДокМ.Количество) КАК Количество,

МАКСИМУМ(ЕСТЬNULL(ОстМ.КоличествоОстаток, 0)) КАК КоличествоОстатков,

МАКСИМУМ(ЕСТЬNULL(ОстМ.СуммаОстаток, 0)) КАК СуммаОстатков

ИЗ

Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки(&МоментВремени, ОтветственныйСотрудник = &ОтвСотр) КАК ОстМ

ПО ДокМ.Номенклатура = ОстМ.Номенклатура

ГДЕ

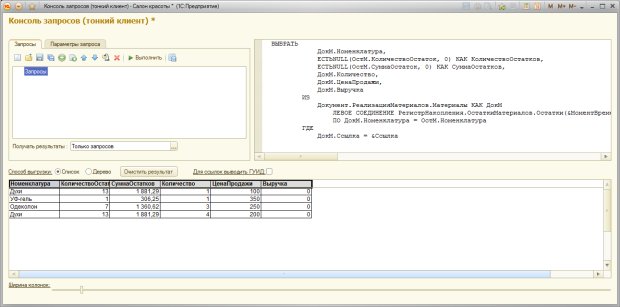
ДокМ.Ссылка = &Ссылка

СГРУППИРОВАТЬ ПО

ДокМ.Номенклатура

В этом запросе нам понадобится откорректировать обращение к таблице документа, который содержит нужные данные. Предполагая, что пользователь может ввести одну и ту же номенклатурную позицию несколькими строками, возможно, собираясь особым образом задать цену продажи для одной и той же позиции, мы не будем группировать запрос по номенклатуре. Мы получим из таблицы все данные, которые мог ввести пользователь – позже мы используем эти данные для заполнения табличного поля.

Остальные данные нас устраивают. У нас получился такой запрос, [рис. 8.3](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=1#image.8.3).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_03.jpg)

**Рис. 8.3.** Запрос для получения данных для заполнения табличного поля

Займемся теперь кодом модуля **РассчитатьНаСервере()**. Воспользуемся конструктором запросов с обработкой результатов для того, чтобы перенести сформированный в консоли текст запроса в код модуля.

В итоге задачу заполнения табличной части данными мы решили следующим образом:

&НаКлиенте

Процедура РассчитатьСебестоимостьИОстатки(Команда)

Режим=РежимДиалогаВопрос.ДаНет;

Ответ=Вопрос("Для продолжения нужно записать документ. Сделать это?",Режим,0);

Если Ответ=КодВозвратаДиалога.Да Тогда

Записать();

РассчитатьНаСервере();

Предупреждение("Табличная часть заполнена");

Иначе

Предупреждение("Табличная часть не заполнена");

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура РассчитатьНаСервере()

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ДокМ.Номенклатура,

| ЕСТЬNULL(ОстМ.КоличествоОстаток, 0) КАК КоличествоОстатков,

| ЕСТЬNULL(ОстМ.СуммаОстаток, 0) КАК СуммаОстатков,

| ДокМ.Количество,

| ДокМ.ЦенаПродажи,

| ДокМ.Выручка

|ИЗ

| Документ.РеализацияМатериалов.Материалы КАК ДокМ

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки(&МоментВремени, ОтветственныйСотрудник = &ОтвСотр) КАК ОстМ

| ПО ДокМ.Номенклатура = ОстМ.Номенклатура

|ГДЕ

| ДокМ.Ссылка = &Ссылка";

Запрос.УстановитьПараметр("МоментВремени", Объект.Ссылка.МоментВремени());

Запрос.УстановитьПараметр("ОтвСотр", Объект.ОтветственныйСотрудник);

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

Результат = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Объект.Материалы.Очистить();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

СтрокаТЧ=Объект.Материалы.Добавить();

СтрокаТЧ.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Если НЕ ВыборкаДетальныеЗаписи.КоличествоОстатков=0 Тогда

СтрокаТЧ.Себестоимость=ВыборкаДетальныеЗаписи.СуммаОстатков/

ВыборкаДетальныеЗаписи.КоличествоОстатков;

КонецЕсли;

СтрокаТЧ.Остаток=ВыборкаДетальныеЗаписи.КоличествоОстатков;

СтрокаТЧ.Количество=ВыборкаДетальныеЗаписи.Количество;

СтрокаТЧ.ЦенаПродажи=ВыборкаДетальныеЗаписи.ЦенаПродажи;

СтрокаТЧ.Выручка=ВыборкаДетальныеЗаписи.Выручка;

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура РассчитатьСумму()

ТекущаяСтрока=Элементы.Материалы.ТекущиеДанные;

ТекущаяСтрока.Выручка=ТекущаяСтрока.Количество\*ТекущаяСтрока.ЦенаПродажи;

КонецПроцедуры

&НаКлиенте

Процедура МатериалыКоличествоПриИзменении(Элемент)

РассчитатьСумму();

КонецПроцедуры

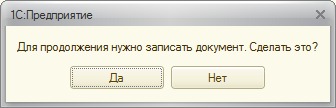
&НаКлиенте

Процедура МатериалыЦенаПродажиПриИзменении(Элемент)

РассчитатьСумму();

КонецПроцедуры

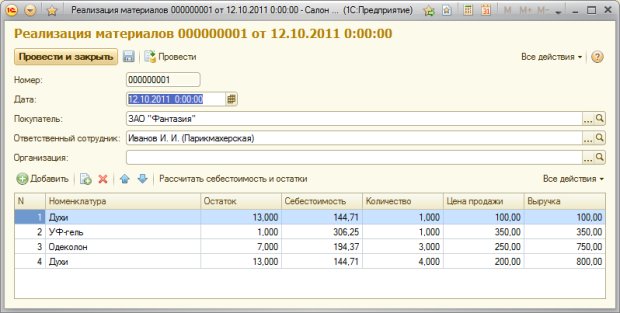
В методе, работающем на клиенте, мы, сразу после запуска, сообщаем пользователю о том, что для правильной работы системы нужно сначала записать документ. Если он отвечает утвердительно – записываем документ и вызываем серверную процедуру. Для задавания подобных вопросов в виде диалоговых окон с кнопками-вариантами вопроса ( [рис. 8.4](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=2#image.8.4).), используется метод **Вопрос()**.



**Рис. 8.4.** Вопрос пользователю

В серверной процедуре мы выбираем с помощью запроса данные из табличной части нашего документа и из регистра **ОстаткиМатериалов**. Когда данные получены, мы просто очищаем табличную часть и заполняем ее снова, теперь уже с использованием новых данных.

В итоге, после того, как пользователь заполнил табличную часть документа и нажал на кнопку **Рассчитать себестоимость и остатки**, он получит примерно следующее, [рис. 8.5](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=2#image.8.5).

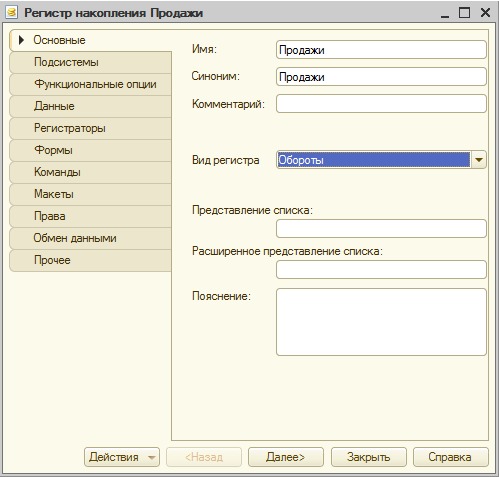
[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_05.jpg)

**Рис. 8.5.** Результат автоматического заполнения табличной части

Поля **Остаток** и **Себестоимость** в данном случае играют лишь вспомогательную роль, позволяя пользователю сразу, при заполнении документа, понять, каково состояние дел с остатками материалов.

Форму документа, да и процедуру заполнения, можно дорабатывать и оптимизировать, но свои основные функции они выполняют, поэтому теперь займемся конструированием оборотного регистра накопления, в котором мы собираемся хранить сведения о продажах. А именно, нас интересуют данные о номенклатуре, о контрагенте, об ответственном лице, продавшем материалы, о количестве, себестоимости и цене материалов.

Создадим новый регистр накопления, назовем его **Продажи**. Вид регистра установим в значение **Обороты**, [рис. 8.6](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=2#image.8.6). Включим его в состав подсистемы **ОперативныйУчетМатериалов**.



**Рис. 8.6.** Создание регистра накопления

В состав данных регистра, [рис. 8.7](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=2#image.8.7)., внесем следующие:

**Измерения**:

**Имя**: Контрагент, тип: СправочникСсылка.Контрагенты

**Имя**: ОтветственныйСотрудник, тип: СправочникСсылка.Сотрудники

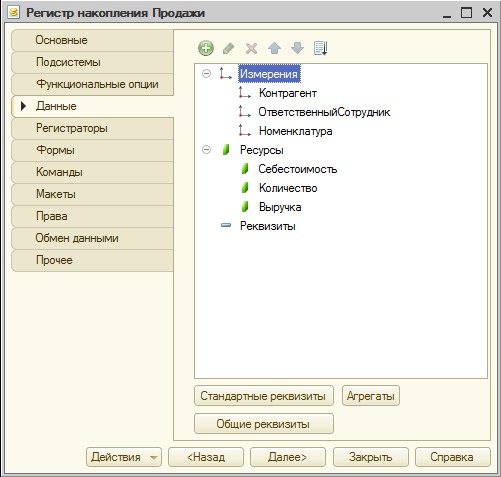
**Имя**: Номенклатура, тип: СправочникСсылка.Номенклатура

Ресурсы:

**Имя**: Себестоимость, тип: Число, длина 10, точность 2

**Имя**: Количество, тип: Число, длина 10, точность 3

**Имя**: Выручка, тип: Число, длина 10, точность 2



**Рис. 8.7.** Настройка данных регистра накопления

В качестве регистратора для данного регистра выберем документ **РеализацияМатериалов**.

Теперь займемся проведением этого документа. Он должен формировать движения по двум регистрам – по регистру **ОстаткиМатериалов**, и по регистру **Продажи**.

Добавим в модуль объекта документа процедуру **ОбработкаПроведения**. При конструировании этой процедуры мы можем воспользоваться уже отработанными при проведении документа **ОтпускМатериаловМастеру** механизмами.

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ДокМ.Номенклатура,

| СУММА(ДокМ.Количество) КАК Количество,

| СУММА(ДокМ.Выручка) КАК Выручка,

| МАКСИМУМ(ЕСТЬNULL(ОстМ.КоличествоОстаток, 0)) КАК КоличествоОстатков,

| МАКСИМУМ(ЕСТЬNULL(ОстМ.СуммаОстаток, 0)) КАК СуммаОстатков

|ИЗ

| Документ.РеализацияМатериалов.Материалы КАК ДокМ

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки(&МоментВремени, ОтветственныйСотрудник = &ОтвСотр) КАК ОстМ

| ПО ДокМ.Номенклатура = ОстМ.Номенклатура

|ГДЕ

| ДокМ.Ссылка = &Ссылка

|

|СГРУППИРОВАТЬ ПО

| ДокМ.Номенклатура";

Запрос.УстановитьПараметр("МоментВремени", МоментВремени());

Запрос.УстановитьПараметр("ОтвСотр", ОтветственныйСотрудник);

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Результат = Запрос.Выполнить();

ВыборкаДЗ = Результат.Выбрать();

Движения.ОстаткиМатериалов.Записывать=Истина;

Движения.Продажи.Записывать=Истина;

Пока ВыборкаДЗ.Следующий() Цикл

Если ВыборкаДЗ.Количество>ВыборкаДЗ.КоличествоОстатков Тогда

Сообщить("Недостаточное количество товара "+ВыборкаДЗ.Номенклатура

+", необходимо "+ВыборкаДЗ.Количество+", в наличии "

+ВыборкаДЗ.КоличествоОстатков);

Отказ=Истина;

Движения.ОстаткиМатериалов.Записывать=Ложь;

Движения.Продажи.Записывать=Ложь;

КонецЕсли;

Если Отказ Тогда

Продолжить;

КонецЕсли;

Движение=Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения=ВидДвиженияНакопления.Расход;

Движение.Период=Дата;

Движение.Номенклатура=ВыборкаДЗ.Номенклатура;

Движение.Количество=ВыборкаДЗ.Количество;

Движение.Сумма=ВыборкаДЗ.Количество\*ВыборкаДЗ.СуммаОстатков/ВыборкаДЗ.КоличествоОстатков;

Движение.ОтветственныйСотрудник=ОтветственныйСотрудник;

Движение=Движения.Продажи.Добавить();

Движение.Период=Дата;

Движение.Номенклатура=ВыборкаДЗ.Номенклатура;

Движение.Количество=ВыборкаДЗ.Количество;

Движение.Себестоимость=ВыборкаДЗ.Количество\*ВыборкаДЗ.СуммаОстатков/ВыборкаДЗ.КоличествоОстатков;

Движение.Выручка=ВыборкаДЗ.Выручка;

Движение.ОтветственныйСотрудник=ОтветственныйСотрудник;

Движение.Контрагент=Покупатель;

КонецЦикла;

КонецПроцедуры

Так, здесь мы, во-первых, проверим достаточность материалов для списания – ранее заполненная табличная часть, служит лишь подсказкой пользователю, к тому же, он может, в процессе работы, получать сведения об остатках и себестоимости единицы, а может и не получать, оставляя соответствующие поля табличной части пустыми, поэтому при проведении документа мы получим с помощью запроса нужные данные из базы.

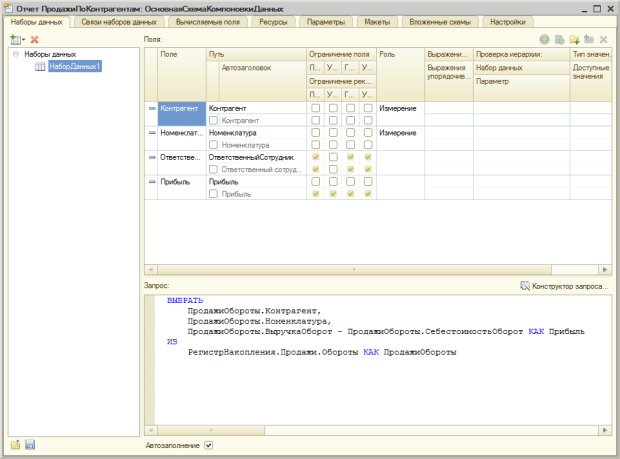
Дальше все идет по уже знакомому вам плану – мы сверяем количество материалов, которое пользователь хочет продать с количеством остатков и принимаем решение либо о продолжении работы, либо – об отмене процедуры проведения документа.

После того, как у нас есть хранилище данных о продажах, мы построим соответствующий отчет. На базе тех данных, которые у нас есть, можно построить различные отчеты – все зависит от того, какие именно данные нас интересуют. Предположим, мы заинтересованы в сведениях о продажах по контрагентам. Нас интересует количественный показатель продажи номенклатурной позиции и прибыль от продажи.

### Отчет о продажах по контрагентам

Создадим новый отчет, дадим ему имя **ПродажиПоКонтрагентам**, добавим его в подсистему **ОперативныйУчетМатериалов**.

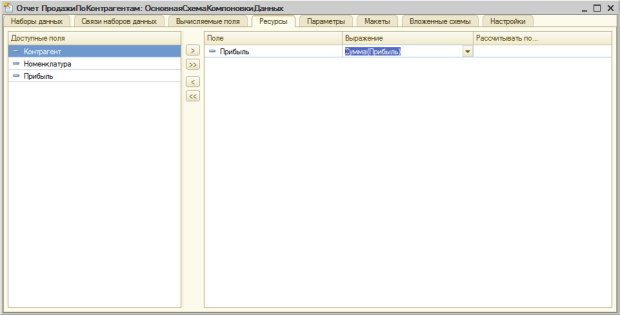
Откроем основную схему компоновки данных отчета. Добавим новый набор данных – **Запрос**. Создадим с помощью конструктора запроса запрос следующего содержания, [рис. 8.8](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.8).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_08.jpg)

**Рис. 8.8.** Набор данных в конструкторе СКД

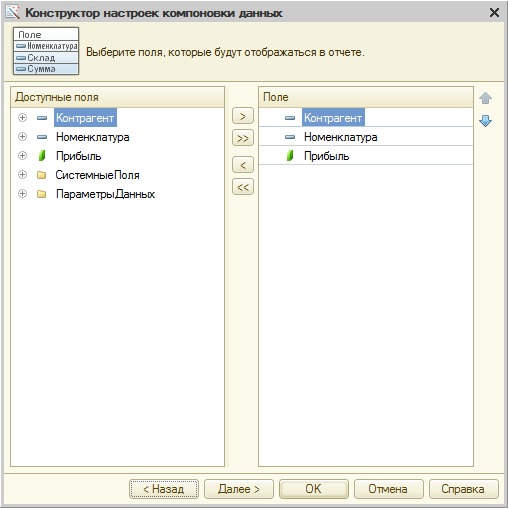
Мы выбираем из виртуальной таблицы **ПродажиОбороты** данные о контрагенте, о количестве проданных материалов и получаем показатель прибыли

На закладке **Ресурсы** мы указываем, что поле **Прибыль** является ресурсом, применяем к нему функцию **Сумма**, [рис. 8.9](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.9).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_09.jpg)

**Рис. 8.9.** Выбор ресурсов в конструкторе СКД

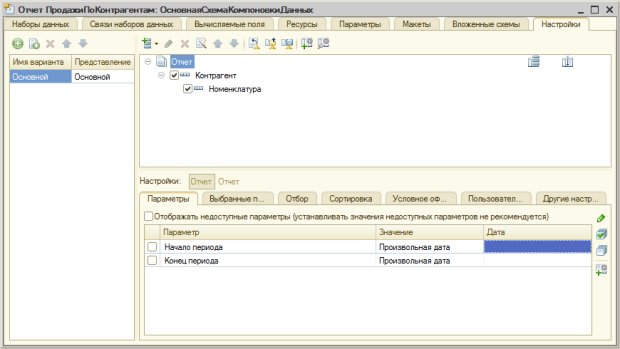
На вкладке **Настройки** мы включим отображение параметров **Начало** периода и **Конец** периода в пользовательском интерфейсе и, вызывав конструктор настроек компоновки данных, выберем тип отчета – **Список**, в качестве полей, которые будут отображаться в отчете, выберем **Контрагент**, **Номенклатура** и **Прибыль**, [рис. 8.10](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.10).



**Рис. 8.10.** Выбор полей отчета при представлении отчета в виде списка

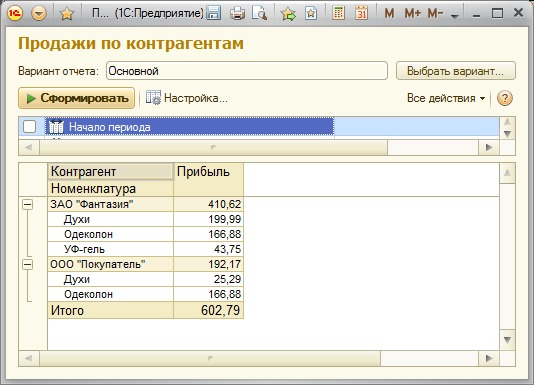
В следующем окне в качестве полей, по которым будет производиться группировка данных, выберем **Контрагент** и **Номенклатура**.

В качестве поля для упорядочивания, в следующем окне конструктора, выберем поле **Контрагент**, упорядочивать будем по возрастанию. После этого нажмем **ОК** и вкладка **Настройки** окна конструктора СКД приобретет такой вид, [рис. 8.11](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.11).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_11.jpg)

**Рис. 8.11.** Вкладка Настройки окна конструктора СКД

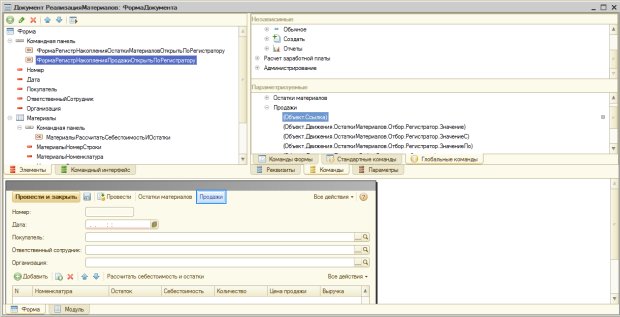
На [рис. 8.12](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.12). представлены результаты работы отчета в пользовательском режиме.



**Рис. 8.12.** Отчет Продажи по контрагентам

### Команда вызова формы регистра в форме документа

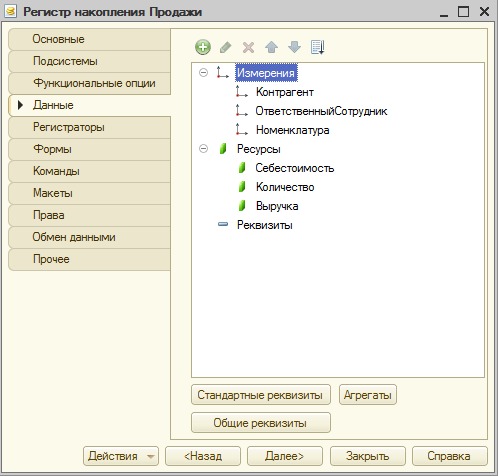
Доработаем форму документа **РеализацияМатериалов**. Документ проводится по двум регистрам накопления – по регистру **Продажи** и по регистру **ОстаткиМатериалов**. Удобно было бы иметь возможность прямо из документа перейти в окно регистра и просмотреть его содержимое после проведения документа. Для того, чтобы реализовать это, нам понадобится закладка **Глобальные команды** закладки **Команды** окна редактора форм. Все, что нужно для размещения на командной панели формы дополнительных кнопок, ведущих к регистрам – перетащить нужную команду на панель, [рис. 8.13](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=3#image.8.13).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_13.jpg)

**Рис. 8.13.** Кнопки перехода в регистры накопления на командной панели формы

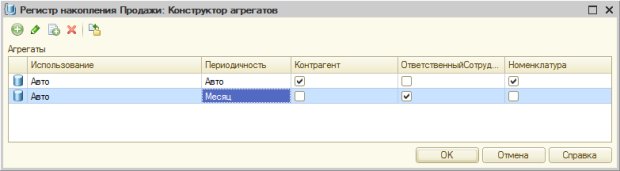
### Агрегаты

Выше мы упоминали о том, что оборотные регистры накопления поддерживают полезный, в плане увеличения быстродействия прикладного решения, механизм агрегатов. **Агрегат** представляет собой хранилище данных, содержащее агрегированную информацию из регистра в различных разрезах. Разрезы подбираются исходя из использования данных регистра в отчетах. Механизм агрегатов позволяет ускорить доступ к информации. Для создания агрегатов используется конструктор агрегатов, который можно вызвать, нажав на кнопку **Агрегаты**, расположенную на закладке **Данные** окна редактирования объекта для оборотного регистра накопления, [рис. 8.14](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.14).



**Рис. 8.14.** Кнопка Агрегаты в окне редактирования объекта

По нажатию на эту кнопку можно открыть окно **Конструктора агрегатов** ( [рис. 8.15](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.15).), где можно добавлять новые агрегаты, настраивать состав измерений регистра, входящих в агрегаты, а так же их свойства. При установленной периодичности **Авто** система сама подбирает подходящую периодичность для агрегата. Мы установили такую периодичность для агрегата, в который входят сведения по измерениям **Контрагент** и **Номенклатура**.



**Рис. 8.15.** Конструктор агрегатов

По измерению **ОтветственныйСотрудник** мы собираемся получать данные ежемесячно, поэтому у агрегата, содержащего указанное измерение, мы установим периодичность **Месяц**.

В нашей учебной базе, содержащей небольшие объемы данных, мы вряд ли сможем оценить полезность агрегатов, однако, при работе с большими базами данных они способны повысить производительность отчетов.

### Последовательности

В нашей конфигурации есть документы, последовательность ввода которых способна влиять на данные, хранимые в системе. Это три документа – **ПоступлениеМатериалов**, **ОтпускМатериаловМастеру** и **РеализацияМатериалов**. Рассмотрим пример.

В некоей конфигурации есть два документа – один фиксирует поступление материалов, второй – списание. Имеется один регистр накопления, в который эти документы делают записи.

Предположим 10.07.11 мы ввели документ поступления материалов (назовем его П1), который отражает принятие к учету 10 единиц материала М1 по цене 10 рублей за единицу.

12.07.11 мы ввели документ списания материалов (назовем его С1), которым списываем 6 единиц материала М1. Очевидно, что при такой последовтельности ввода документов, в том случае, если мы используем метод расчета себестоимости по средней, окажется, что списано 6 единиц материалов М1 на общую сумму 60 рублей, в остатках мы имеем 4 единицы на 40 рублей.

Что произойдет, если, после того, как два вышеописанных документа уже введены в систему, проведены, мы, например, 11.07.11, введем еще один приходный документ (П2), которым принимаем к учету 10 единиц материала М1 по цене 15 рублей за единицу?

Если мы построим отчет по остаткам материалов, скажем, на 13.07.11, то окажется, что количественные показатели нас устраивают – в остатке будет 14 единиц материала. Стоимость этих материалов будет складываться из стоимости остатка – это 4\*10 и стоимости материалов, принятых задним числом, то есть – 10\*15. Сумма равна 190 рублей.

Документ С1, которым материалы списаны 12.07.11, был проведен без наличия сведений о материалах, поступление которых отражено 11.07.11. А это значит, что при применении метода расчета себестоимости по средней, этот документ содержит неверные данные. Действительно, на момент списания материалов этим документом их количество равнялось 10, общая стоимость 100, средняя стоимость составила 10. После ввода документа П2 мы имеем на 12.07.11 уже 20 единиц материала на сумму 250 рублей, средняя стоимость единицы уже равняется 12,5 рублей. То есть, наш документ С1 должен списать 6 единиц материалов М1 по 12,5 рублей, что дает нам их общую стоимость в 75 рублей и приводит к тому, что в остатках на 13.07.11 должны числиться 14 единиц, себестоимость которых должна равняться 175 рублей.

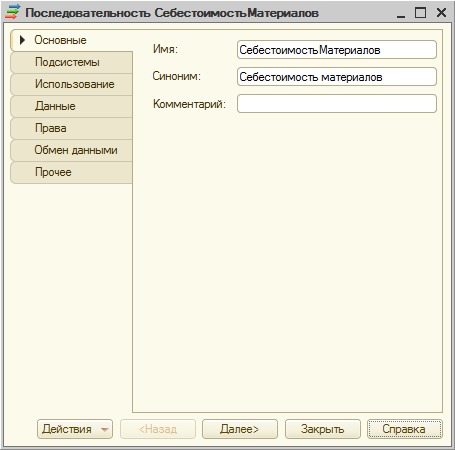
Если при проведении документа С1 данные по документу М2 не учитывались – мы получаем в остатке 14 единиц, себестоимость которых равняется 190 рублей. Все это значит, что документ, введенный задним числом, искажает данные учета. Если абстрагироваться от нашей учебной конфигурации и рассмотреть последствия подобной ситуации в реальной учетной системе, то окажется, что такое происшествие способно исказить данные учета, как результат – повлиять на суммы налогов (например, налога на прибыль). То есть, даже если не говорить о том, что в учете будут отражены некорректные данные, подобная "ошибка" способна привести к весьма неприятным для организации последствиям в виде санкций налоговых органов.

Для того, чтобы подобную ситуацию разрешить, нужно, как вы уже, наверное, догадались, после ввода "дополнительного" документа П2 просто перепровести документ С1. Все это выглядит простым и понятным в примере с тремя документами, но в учетной системе с десятком видов документов, несколькими регистрами, в которую вводят несколько сотен различных документов ежедневно, "вручную" отследить возникновение подобной ситуации достаточно сложно. Можно, конечно, прибегнуть к радикальному методу – например, периодически перепроводить все документы за какой-либо период или за все время работы в базе. Но перепроведение большого количества документов – дело трудоемкое, к тому же, если уж полностью следовать логике полного перепроведения, перепроводить нужно все документы с начала работы в базе. А это, возможно, документы, накопленные за несколько лет, то есть – операция слишком ресурсозатратная для ее регулярного выполнения.

К счастью, разработчики 1С:Предприятие 8.2. предусмотрели возникновение подобных ситуаций – для контроля за правильной последовательностью проведения документов в системе имеется объект, который так и называется – **Последовательность**.

Последовательность позволяет автоматически контролировать хронологическую последовательность проведения документов, включенных в нее, а так же – для документов, влияющих на состояние отслеживаемых последовательностью регистров.

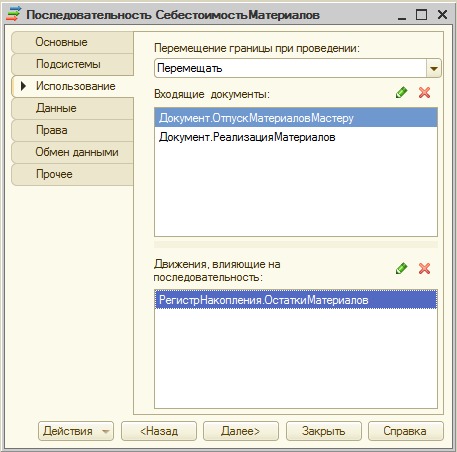
Создадим новую последовательность (ветвь дерева конфигурации **Последовательности** находится в ветви **Документы**), назовем ее **СебестоимостьМатериалов**, [рис. 8.16](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.16).



**Рис. 8.16.** Создание новой последовательности

Включим последовательность **СебестоимостьМатериалов** в подсистему **ОперативныйУчетМатериалов**.

На закладке **Использование**, [рис. 8.17](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.17)., мы оставим параметр **Перемещение границы при проведении** в значении **Перемещать**, в поле **Входящие документы** внесем документы **ОтпускМатериаловМастеру** и **РеализацияМатериалов**, в поле **Движения, влияющие на последовательность**, внесем регистр накопления **ОстаткиМатериалов**.

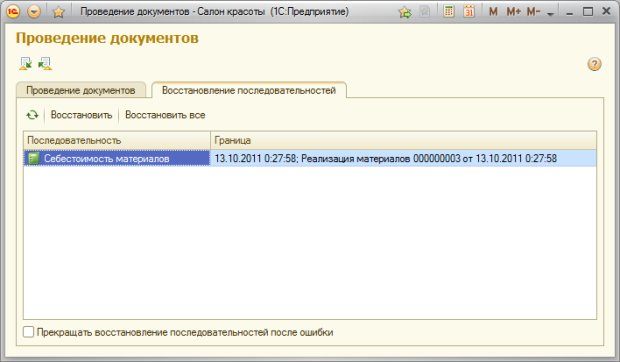


**Рис. 8.17.** Настройка последовательности

Сущность действия последовательности заключается в следующем. Она отслеживает документы, влияющие на нее (и документы, влияющие на состояние регистра, в нашем случае это документ **ПоступлениеМатериалов**) и, в зависимости от даты проведения документов перемещает границу либо вперед (при нормальной хронологической последовательности ввода документов), либо – назад – в том случае, если документ введен (проведен) не в нормальном порядке.

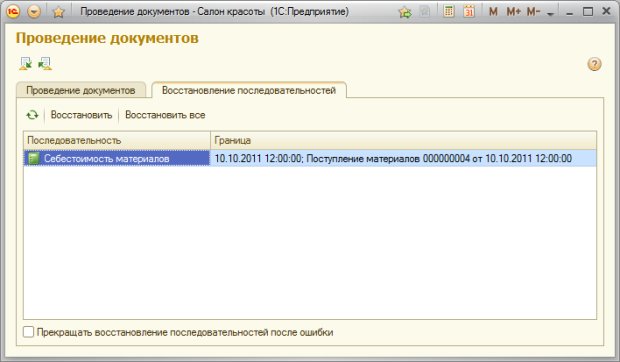
Граница последовательности указывает нам на документ, документы, введенные после которого могут отражать некорректные данные. Последовательность можно восстанавливать – при восстановлении последовательности перепроводятся документы, входящие в нее. В нашем случае это документы, которыми мы списываем материалы. Документы поступления материалов влияют на последовательность, но они вводят в систему исходные данные для расчета себестоимости, поэтому внеочередное проведение документа поступления материалов на другие документы поступления не влияет, но вполне может повлиять на документы списания материалов.

При восстановлении последовательности производится автоматическое проведение нужных документов и граница последовательности устанавливается на последний из них. Для управления последовательностями в пользовательском режиме можно выполнить команду **Главное меню > Все функции > Стандартные > Проведение документов**, и там, на закладке Последовательности, [рис. 8.18](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.18)., посмотреть состояние последовательностей, и, при необходимости, восстановить их.



**Рис. 8.18.** Работа с последовательностями

Здесь мы видим, что граница последовательности установлена на документ **Реализация материалов** от 13.10.2011. Введем документ **ПоступлениеМатериалов**, например, от 10.10.2011. Это приведет к такому состоянию последовательности, [рис. 8.19](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=4#image.8.19).

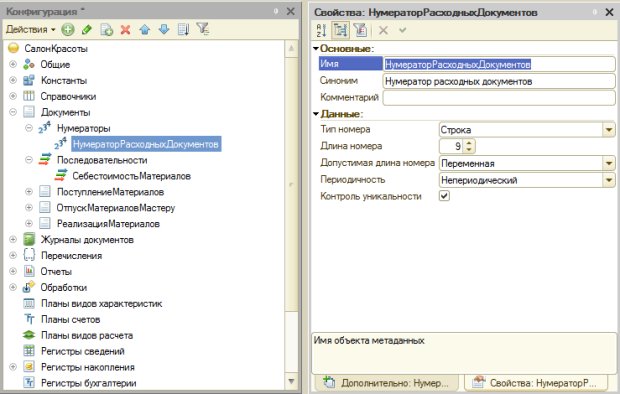


**Рис. 8.19.** Перемещение границы последовательности при вводе документа в прошлом периоде

Восстановление последовательности приведет к перепроведению документов, входящих в нее и снова установит ее границу на документ Реализация материалов от 13.10.2011.

### Нумераторы

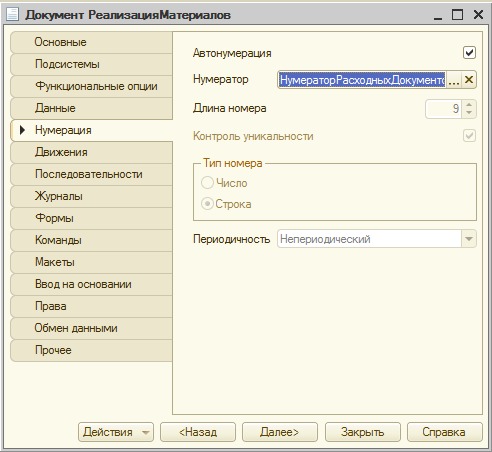
Еще один полезный объект, который можно найти в ветви **Документы** дерева конфигурации – это **нумераторы**. Нумератор позволяет задавать единые правила нумерации для различных документов. Создадим новый нумератор, назовем его **НумераторРасходныхДокументов**, [рис. 8.20](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=5#image.8.20).



**Рис. 8.20.** Создание нового нумератора

Редактирование свойств нумератора осуществляется в окне свойств объекта, для него не предусмотрено окна редактирования объекта.

Благодаря нумератору документы разных видов, которым он назначен (делается это на вкладке **Нумерация** окна настройки свойств документа), [рис. 8.21](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=5#image.8.21). приобретают сквозную нумерацию.



**Рис. 8.21.** Настройка использования нумератора в документе

То есть, если, например, мы создали документ РеализацияМатериалов и он приобрел номер 000000001, то, если следующим документом с тем же нумератором будет документ ОтпускМатериаловМастеру, ему автоматически будет присвоен номер 000000002.

### Регистры сведений

Мы хотели бы хранить и использовать в нашей конфигурации сведения о различных валютах. Решение подобной задачи автоматически заставляет нас сначала задуматься о каталоге валют в конфигурации, то есть – о соответствующем справочнике. Мы можем создать такой справочник, оснастить его реквизитами, необходимыми для хранения сведений о валютах. Предположим, мы решим хранить данные о курсе валюты так же, в справочнике. Мы можем это сделать, но такой подход будет не слишком удобным.

Главное неудобство здесь заключается в том, что, введя в карточку валюты сведения о курсе на определенную дату, мы потеряем предыдущее значение курса. Справочник не хранит историю значений реквизитов, поэтому мы вынуждены будем либо довольствоваться единственным значением курса, либо использовать для хранения курсов валют какой-то другой механизм. Такой механизм имеется в 1С:Предприятие 8 и называется он регистром сведений.

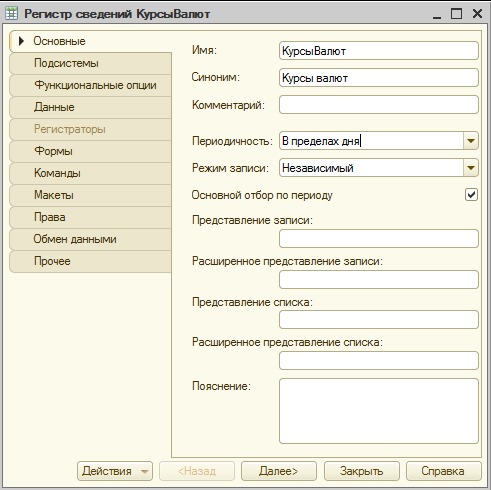
Регистры сведений бывают различных видов. В первом приближении они делятся на периодические и непериодические регистры. Непериодические регистры проще всего сравнить с некими таблицами, в которых хранятся определенные сведения, развернутые по заданным в регистре измерениям. Примерами таких регистров могут быть регистр, фиксирующий содержание драгоценных металлов в каких-либо деталях, регистр, хранящий данные адресного классификатора, регистр с информацией о том, какой день недели является рабочим, а какой – выходным, и многие другие. Даже если одним из измерений непериодического регистра сведений является дата – он не становится от этого периодическим.

В противоположность непериодическому регистру, который, в сущности, хранит одни и те же показатели, периодический регистр может хранить историю изменения этих показателей. Такие регистры могут иметь различную периодичность. Например, для регистра, призванного фиксировать курсы валют, разумно будет установить периодичность в пределах дня, так как курсы валют обычно фиксируются в пределах одного дня.

Другая классификация регистров сведений заключается в том, является ли регистр независимым или подчиненным регистратору. Независимый регистр сведений мы можем заполнять самостоятельно, вручную вводя в него данные непосредственно в таблице регистра. Когда же регистр подчинен регистратору, записи в него производятся при проведении документа-регистратора.

Рассмотрим использование периодического независимого регистра сведений на примере с курсами валют. Для начала создадим новый справочник. Назовем его **Валюты**, включим в подсистему **БухгалтерскийУчет**, других настроек для данного справочника задавать не будем. Нас вполне устроит возможность задать наименование валюты с помощью стандартного реквизита.

Создадим новый регистр сведений **КурсыВалют**, установим его периодичность – В пределах дня, [рис. 8.22](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=5#image.8.22).



**Рис. 8.22.** Создание регистра сведений

На закладке **Подсистемы** включим регистр в подсистему **БухгалтерскийУчет**.

На закладке **Данные** ( [рис. 8.23](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=5#image.8.23).) добавим следующее:

**Измерения**:

**Имя**: Валюта, тип СправочникСсылка.Валюты. Флаги Ведущее, Основной отбор и Запрет незаполненных значений установлены

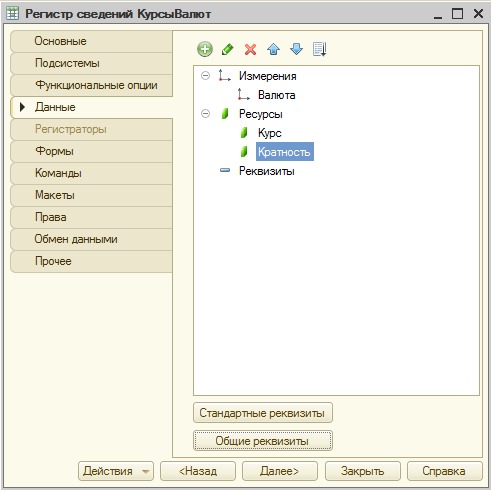
**Ресурсы**

**Имя**: Курс, тип Число, длина 10, точность 4

**Имя**: Кратность, тип Число, длина 10, точность 0

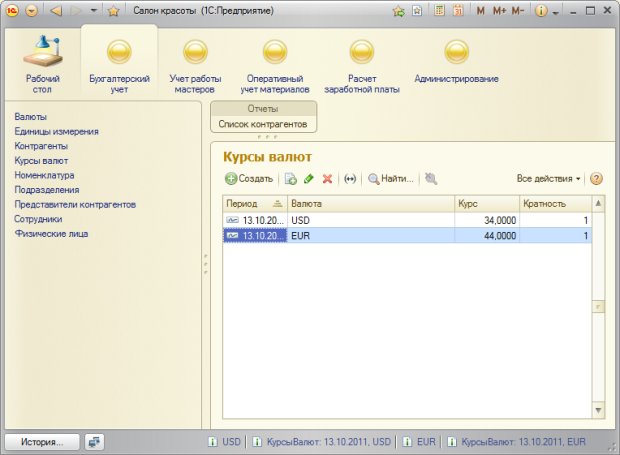
Исключим регистр из состава общего реквизита **Организация**.

В ресурсе **Курс** мы будем хранить курс, который может выражаться числом с точностью до 4-х знаков после запятой. В ресурсе **Кратность** мы будем хранить кратность валюты по отношению к рублю. Например, для американского доллара курс может быть 34.3234, а кратность 1 – то есть, за один доллар дают 34.3234 рубля.



**Рис. 8.23.** Настройка состава данных регистра сведений

Воспользуемся созданной группой объектов в пользовательском режиме. Здесь мы можем вручную вводить данные по валютам, [рис. 8.24](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=5#image.8.24).



**Рис. 8.24.** Заполнение регистра сведений в пользовательском режиме

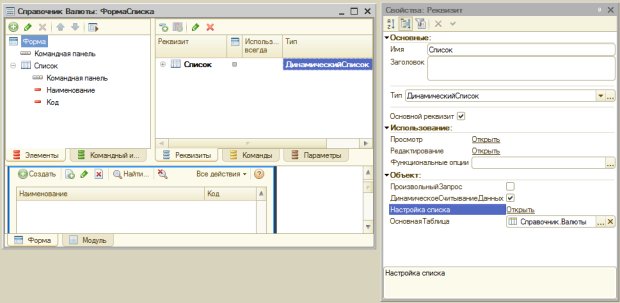
Теперь в нашей конфигурации есть сведения о курсах валют.

### Программное получение данных из регистра сведений

Решим следующую задачу. Нам хотелось бы вывести курс валюты на текущую дату в списке справочника **Валюты**.

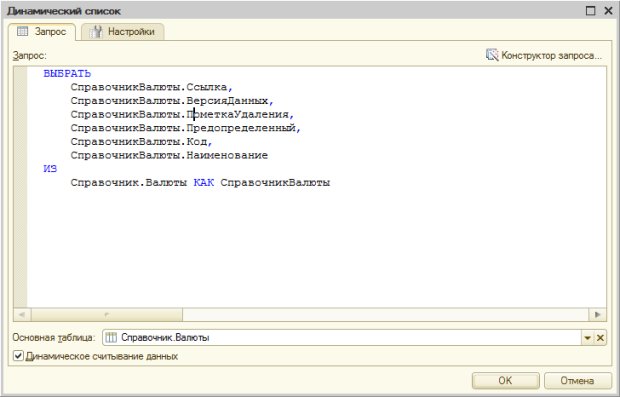
Для начала создадим форму списка справочника **Валюты**.

Обратите внимание на то, что реквизит **Список** имеет тип **ДинамическийСписок**. Это означает, что мы можем вмешаться в создание этого списка, так как он строится на основе некоего запроса, генерируемого, в данном случае, системой автоматически. Для того, чтобы самостоятельно отредактировать запрос, который лежит в основе динамического списка, нам нужно в окне его свойств ( [рис. 8.25](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=6#image.8.25).) установить флаг **ПроизвольныйЗапрос**.

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_25.jpg)

**Рис. 8.25.** Редактирование свойств динамического списка

После установки этого флага мы можем нажать на ссылку **Открыть** у поля **Настройка списка** и увидеть, какой именно запрос система построила для создания этого списка, [рис. 8.26](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=6#image.8.26).



**Рис. 8.26.** Запрос по умолчанию для заполнения динамического списка

Дополним запрос таким образом, чтобы он выводил в дополнение к запрошенным полям еще и наиболее свежее значение курса. Запрос можно ввести вручную в поле Запрос, воспользоваться конструктором запроса, доступным из этого же поля или, как мы уже делали, сначала отладить запрос в консоли запросов, а потом добавить его в поле **Запрос**. То, что мы хотим, можно сделать с помощью следующего запроса:

ВЫБРАТЬ

СправочникВалюты.Ссылка,

СправочникВалюты.ВерсияДанных,

СправочникВалюты.ПометкаУдаления,

СправочникВалюты.Предопределенный,

СправочникВалюты.Код,

СправочникВалюты.Наименование,

КурсыВалютСрезПоследних.Курс

ИЗ

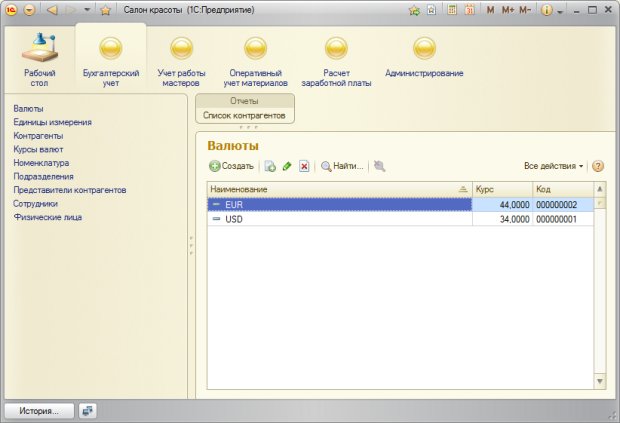
Справочник.Валюты КАК СправочникВалюты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КурсыВалют.СрезПоследних КАК КурсыВалютСрезПоследних

ПО СправочникВалюты.Ссылка = КурсыВалютСрезПоследних.Валюта

Здесь мы получаем из виртуальной таблицы **КурсыВалютСрезПоследних** наиболее поздние значения курса, не прибегая к специальной настройке ее параметров.

После показанной модификации запроса, формирующего динамический список и размещения в форме списка справочника нового поля **Курс**, которое будет доступно в списке реквизитов динамического списка, форма списка приобретет вид, показанный на [рис. 8.27](https://www.intuit.ru/studies/courses/2318/618/lecture/13478?page=6#image.8.27).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/8/files/08_27.jpg)

**Рис. 8.27.** Измененная форма списка справочника