

Лабораторная работа № 15

РАЗРАБОТКА КОНФИГУРАЦИИ ДЛЯ УЧЕТА ДОХОДОВ ОТ ПРОДАЖ ТОВАРОВ

Сложность: **

Теги: справочник, документ, функциональная опция, схема компоновки данных, регистр накопления, регистр сведений

ЗАДАНИЕ

Заказчик просит разработать конфигурацию для учета продаж товаров с сопутствующими услугами покупателям. Необходимо предусмотреть опциональную возможность использования различных валют.

При мультивалютном учете пользователь системы при оформлении продажи должен обязательно указать валюту. Итоговая стоимость заказа должна формироваться автоматически.

В системе нужно реализовать хранилище суммы доходов в рублях по номенклатурным позициям.

Доходом считается сумма продажи в рублевом выражении.

Необходимо построить «Отчет по продажам» с упорядочиванием по сумме доходов.

Форма отчета:

Номенклатура	Сумма
Компьютер	239 000,00
Телефон	45 000,00
Доставка	34 000,00
Итого	318 000,00

Отчет выводит информацию по выбранной валюте, а также подводит общий итог.

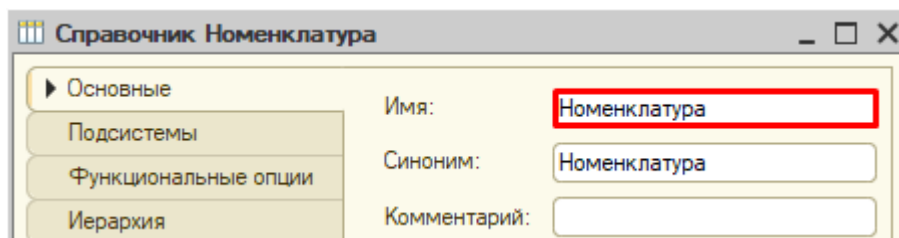
Выполнение

Из условия следует, что нужно хранить информацию о покупателях, товарах и услугах, а также валютах. Для решения этой задачи нам понадобятся справочники.

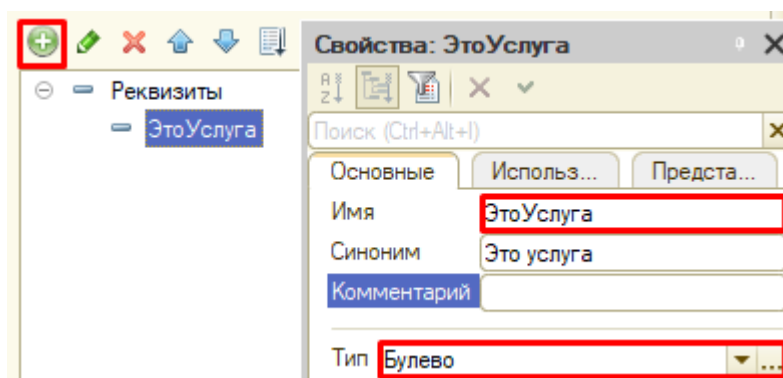
Определение

Справочник – это объект конфигурации, который хранит справочную информацию, например, перечень товаров или список сотрудников (более подробно про справочники можно прочитать здесь: <https://v8.1c.ru/platforma/spravochniki/>).

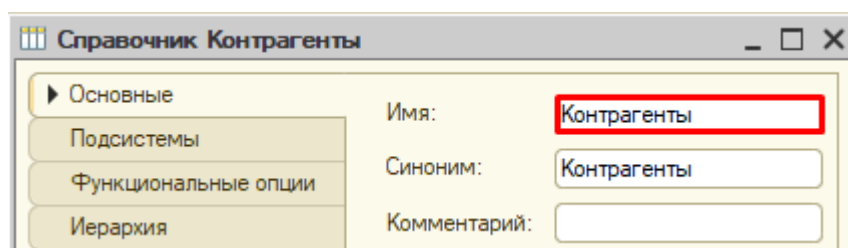
Создадим справочник «Номенклатура».



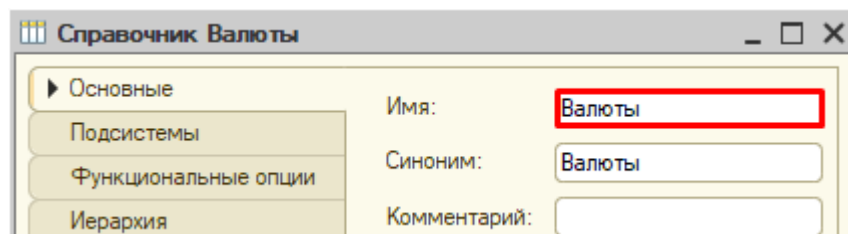
Перейдем на вкладку «Данные» и добавим реквизит «ЭтоУслуга», тип – «Булево». Данный реквизит необходим для того, чтобы отличать товары от услуг в справочнике.



Создадим справочник «Контрагенты».



Создадим справочник «Валюты».

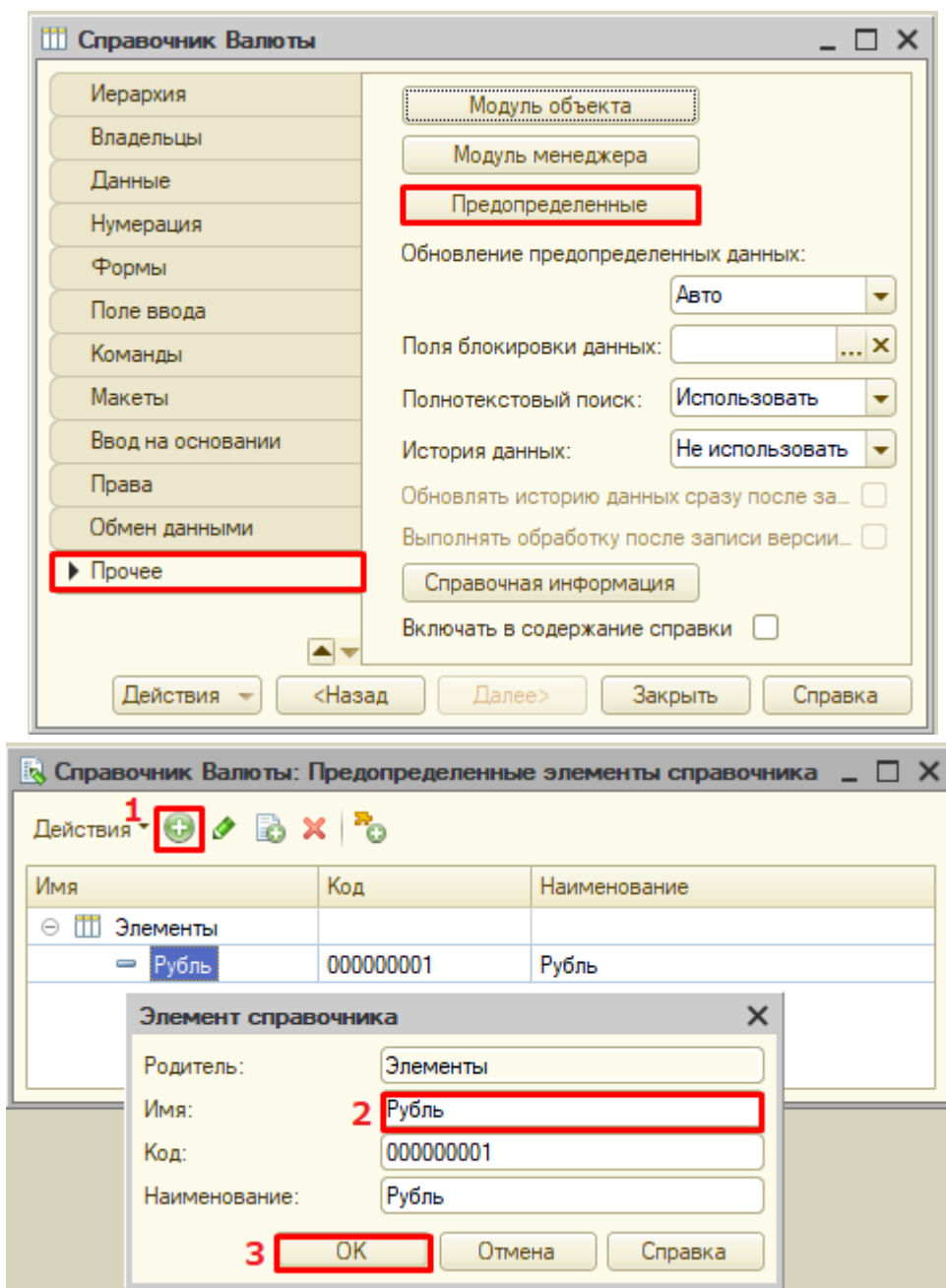


Обязательной валютой в данной информационной системе является рубль. Создадим predetermined element.

Определение

Предопределенные элементы – это такие элементы, которые создает разработчик в конфигураторе для удобства работы пользователя.

Созданный таким образом элемент будет доступен пользователю с первого запуска программы. Для создания такого элемента перейдем на вкладку «Прочее» и откроем список предопределенных элементов справочника.



Откроем программу в режиме «1С:Предприятие» и добавим в каждый справочник несколько элементов.

← → ☆ **Контрагенты** 🔗 ⋮ ×

Создать 📄 Поиск (Ctrl+F) × Еще ▾

Наименование	↓	Код

Контрагенты (создание) ⋮ □ ×

Записать и закрыть Записать Еще ▾

Код:

Наименование:

← → ☆ **Контрагенты** 🔗 ⋮ ×

Создать 📄 Поиск (Ctrl+F) × Еще ▾

Наименование	↓	Код
— ООО "Василёк"		000000001
— ООО "Мак"		000000002

Аналогично добавьте элементы в справочники «Номенклатура» и «Валюты».

← → ☆ **Номенклатура** 🔗 ⋮ ×

Создать 📄 Поиск (Ctrl+F) × 🔍 Еще ▾

Наименование	↓	Код	Это услуга
— Доставка		000000003	✓
— Компьютер		000000002	
— Настройка		000000004	✓
— Телефон		000000001	

← → ☆ **Валюты** 🔗 ⋮ ×

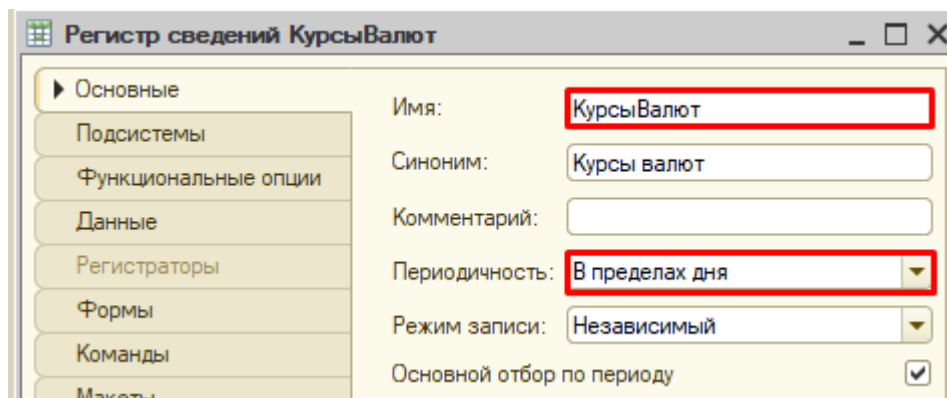
Создать 📄 Поиск (Ctrl+F) × 🔍 Еще ▾

Наименование	↓	Код
— Доллар		000000002
— Рубль		000000001

Заметим, что predetermined элементы в справочнике отмечены желтым маркером.

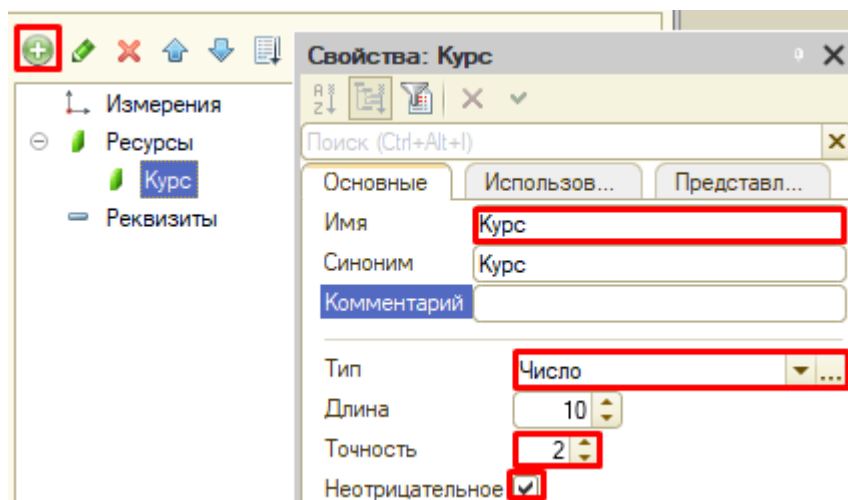
Чтобы хранить курсы валют, которые меняются со временем, нам понадобится специально настроенный справочник. Такой прикладной объект называется *регистр сведений* (подробнее про *регистры сведений* можно прочитать здесь: <https://v8.1c.ru/platforma/registr-svedeniy/>).

Создадим *регистр сведений* и назовем его «КурсыВалют». Поскольку курс валют может меняться раз в день, то установим периодичность в пределах дня.

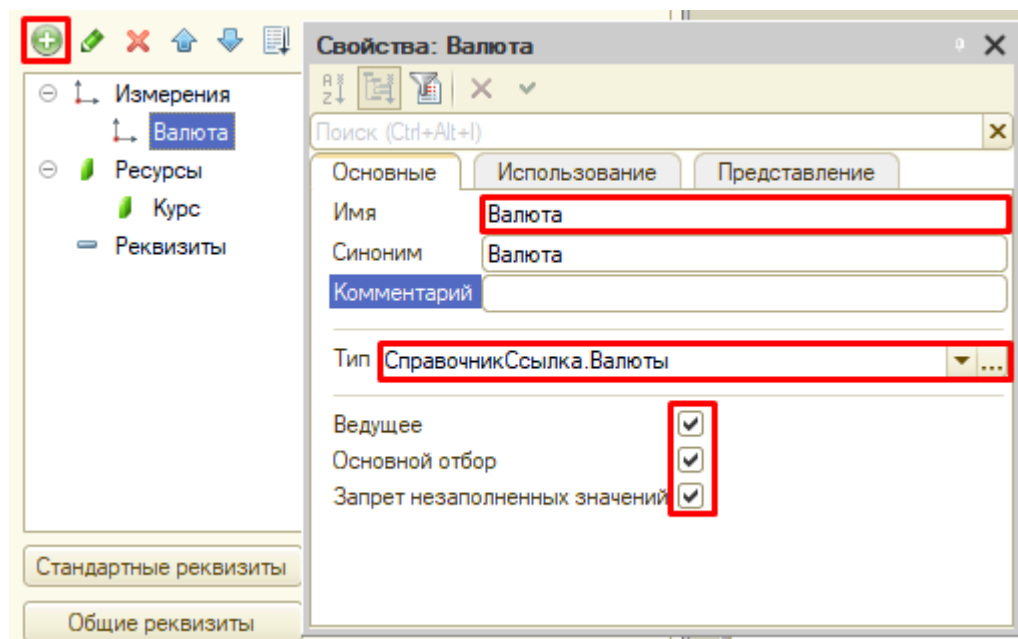


На вкладке «Данные» определим структуру нашего регистра.

Мы должны задаться вопросом: «Что мы хотим хранить?». Мы хотим хранить курс. То, что мы хотим хранить, – это и есть ресурс. Добавим ресурс «Курс».



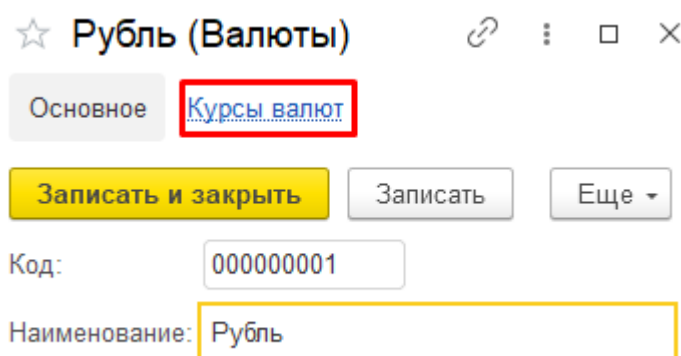
После того, как мы определили, что хотим хранить, нужно понять: в разрезе чего мы хотим это хранить? Курсы мы хотим хранить в разрезе валют. Значит, валюта – измерение.



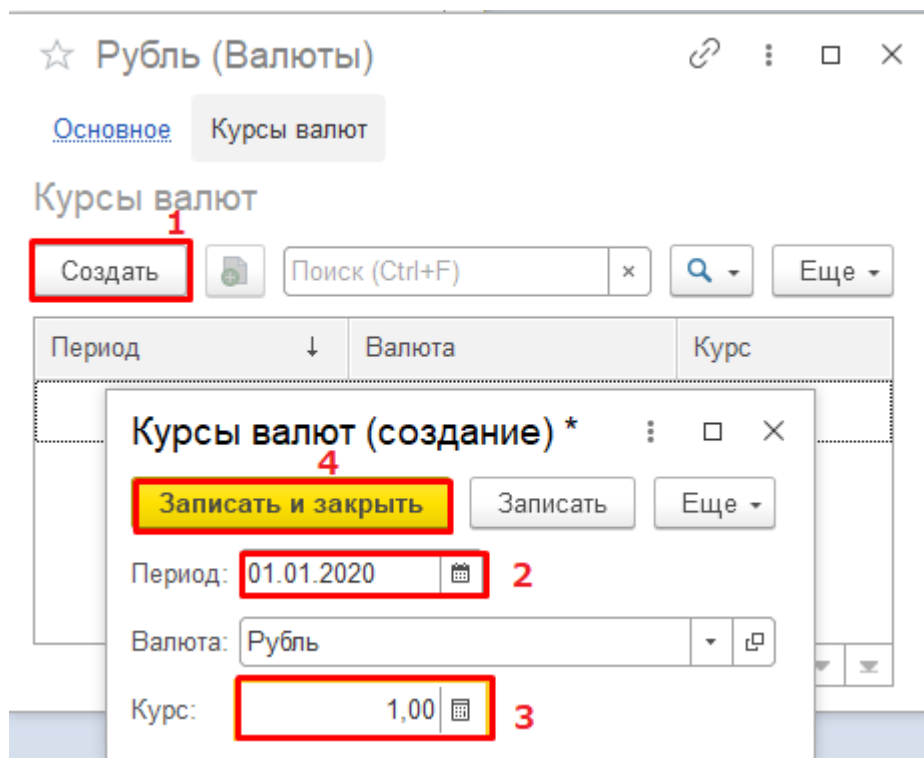
Свойство «Ведущее» позволит нам переходить к курсу валюты прямо из справочника соответствующего элемента.

Запустим режим «1С:Предприятие» и заполним курсы валют на разные даты.

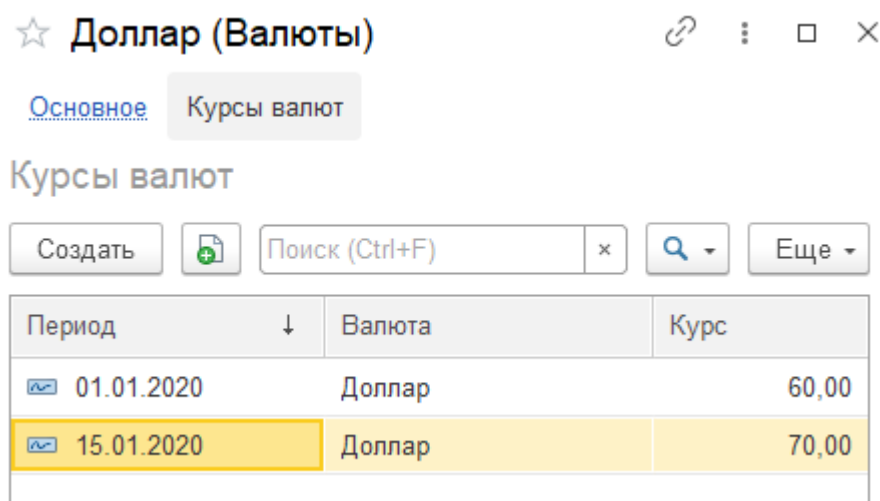
Откроем карточку элемента справочника «Валюты» и убедимся в работоспособности настройки свойства «Ведущее».



Обязательно укажем курс для рубля. Поскольку рубль к рублю идет в отношении 1:1, то укажем курс рубля – «1». Указывать будем на начало года.



Добавьте курсы на разные даты, в том числе и для других валют.

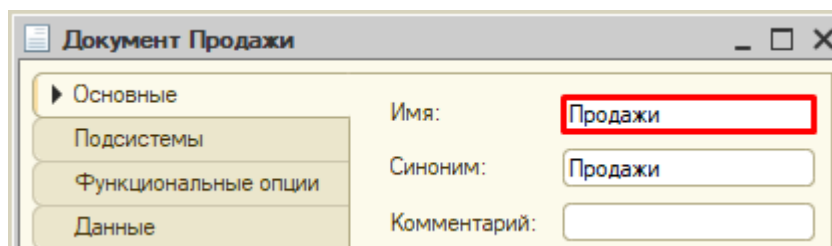


Для регистрации продаж следует воспользоваться объектом конфигурации *документ*.

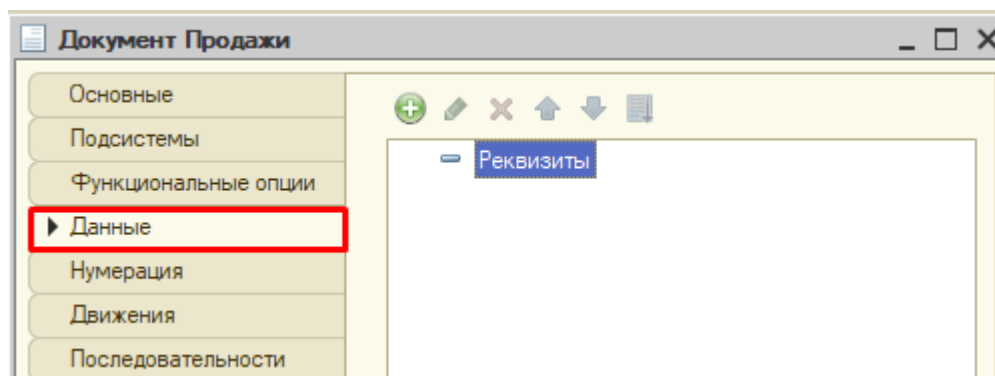
Определение

Документ – это объект конфигурации, хранящий информацию о каких-либо событиях, произошедших в «жизни» предприятия. Например, с помощью документа можно зарегистрировать (то есть сохранить данные документа для последующей обработки) продажу товаров или начисление зарплаты (подробнее про документы можно прочитать здесь: <https://v8.1c.ru/platforma/dokumenty/>).

Добавим новый документ «Продажи».

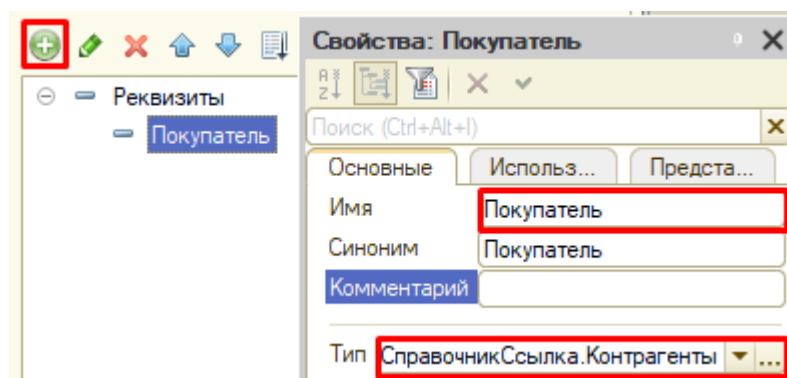


Для настройки структуры документа переходим на вкладку «Данные».

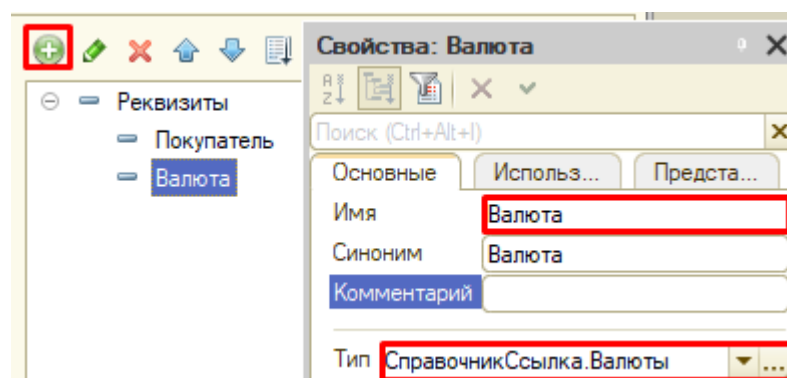


Из условия следует, что при продаже нужно указывать покупателя и валюту. Также будем хранить и итоговую стоимость.

Добавим реквизит «Покупатель».

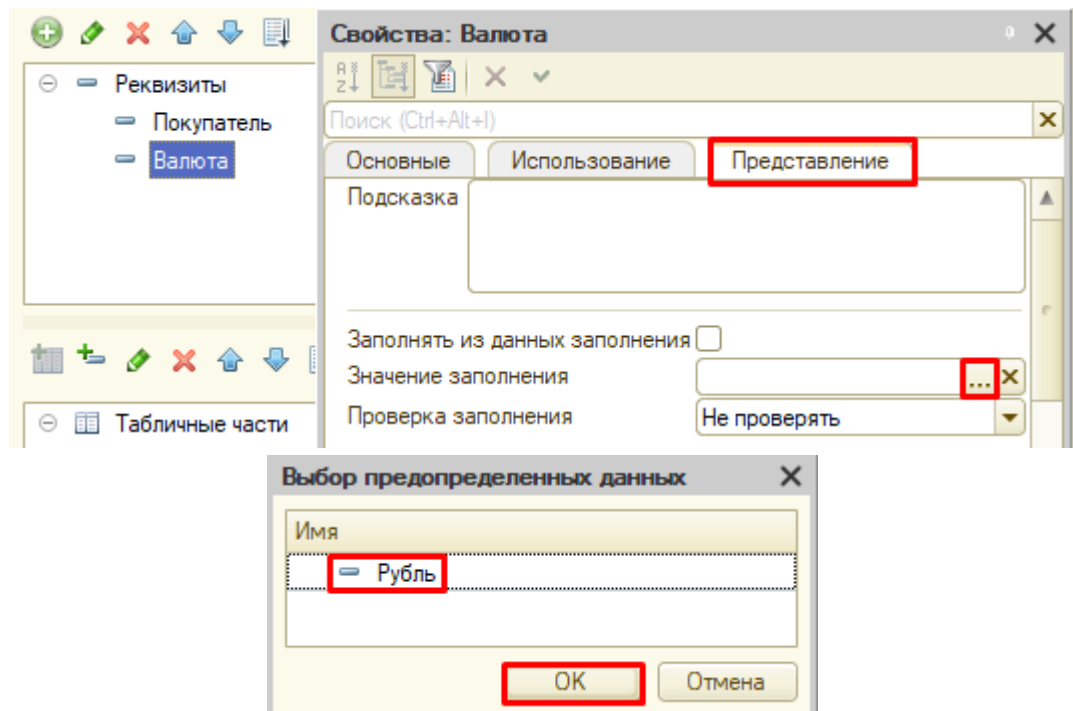


Далее добавим реквизит «Валюта».



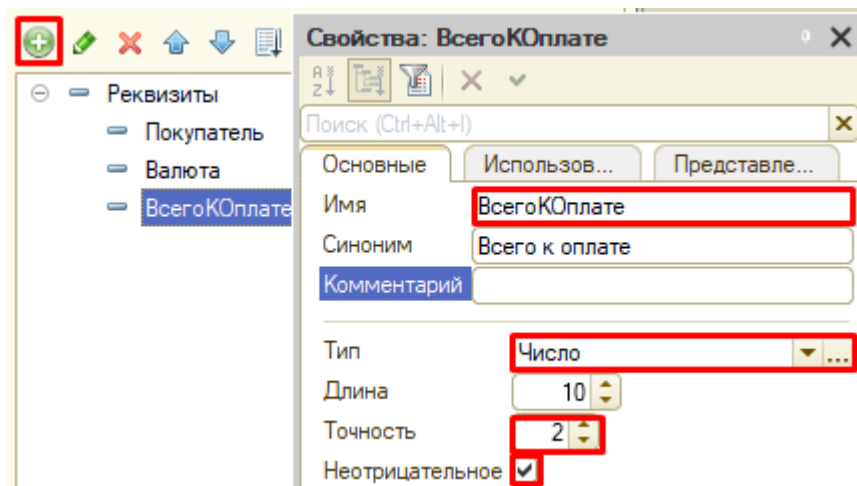
Пользователь может не использовать возможность ведения учета в различных валютах. В этом случае поле «Валюта» у него будет отсутствовать (этот функционал мы реализуем далее), но системе необходимо понимать, к чему привязана итоговая стоимость: к рублям или, например, долларам.

Поэтому установим значения заполнения по умолчанию на вкладке «Представление».



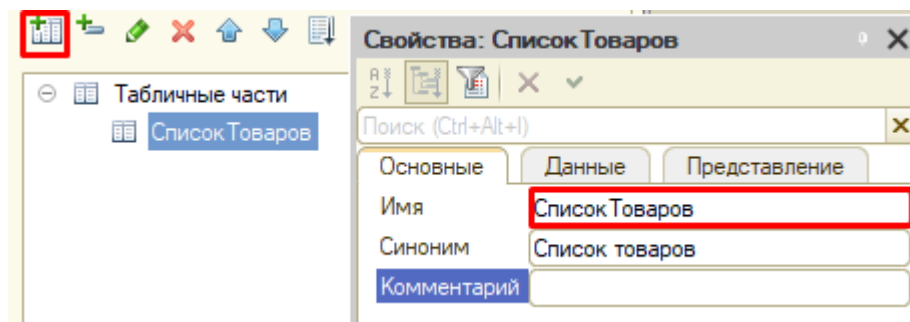
Таким образом, у пользователя по умолчанию будет указана валюта «Рубль», причем пользователь об этом может даже не знать.

Последним реквизитом в шапку документа добавим «ВсегоКОплате».

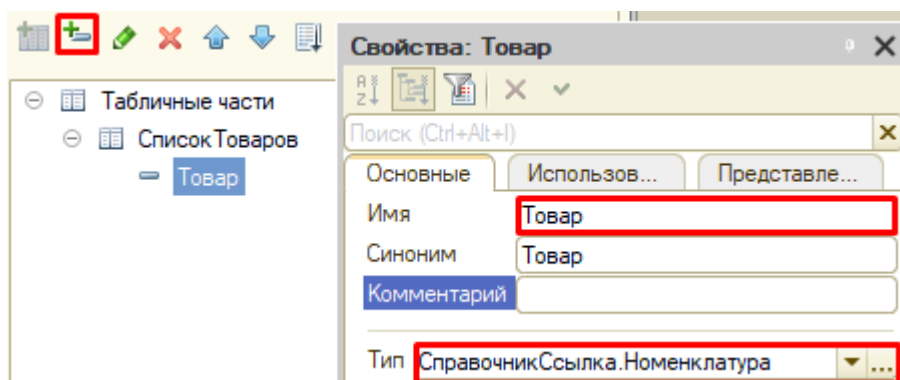


Чтобы регистрировать продажу товаров и услуг в одном документе, необходимо добавить две табличные части.

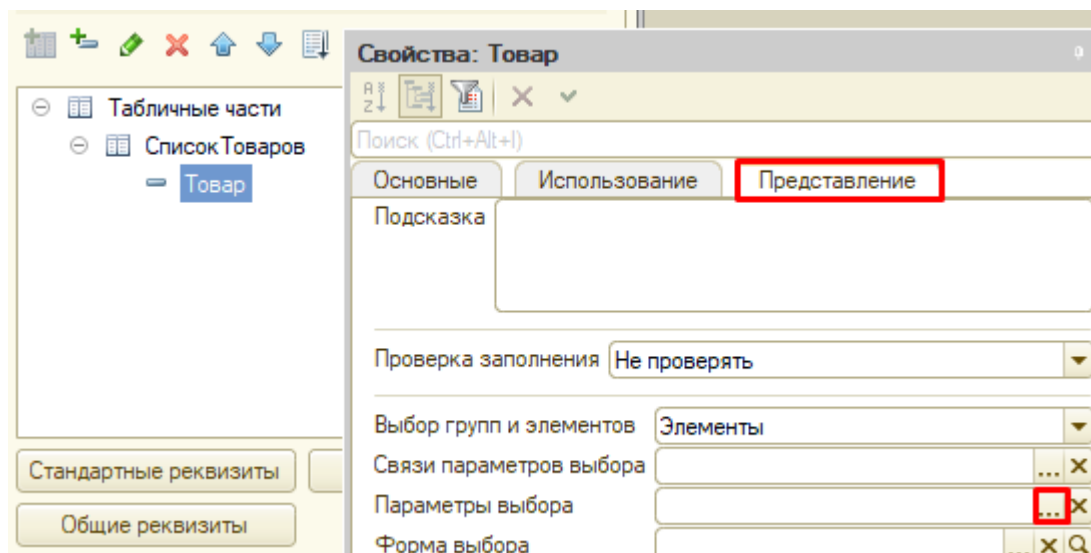
Сначала добавим табличную часть «СписокТоваров».

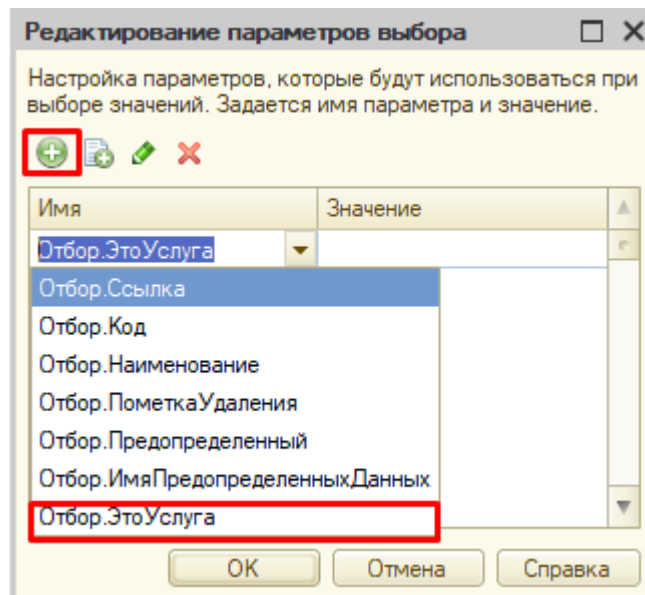


Добавим реквизит табличной части (колонку) «Товар».

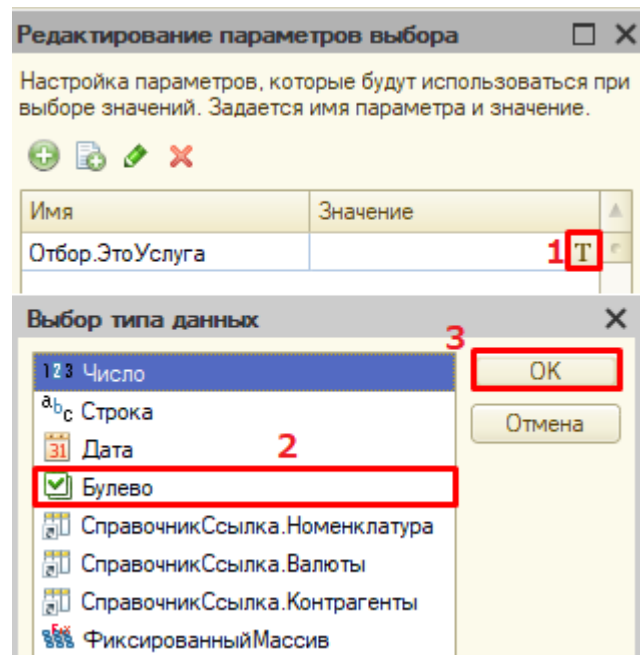


Поскольку мы ссылаемся на общий справочник «Номенклатура», то нам нужно на вкладке «Представление» настроить отбор только по тем элементам, у которых значение реквизита «ЭтоУслуга» установлено в положение «Ложь». Таким образом, пользователь будет в списке выбора видеть только товары.

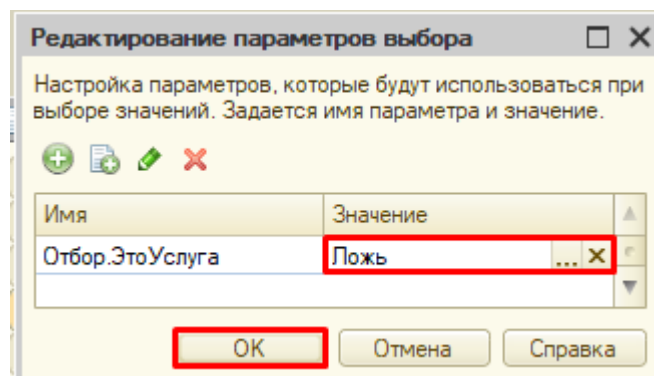




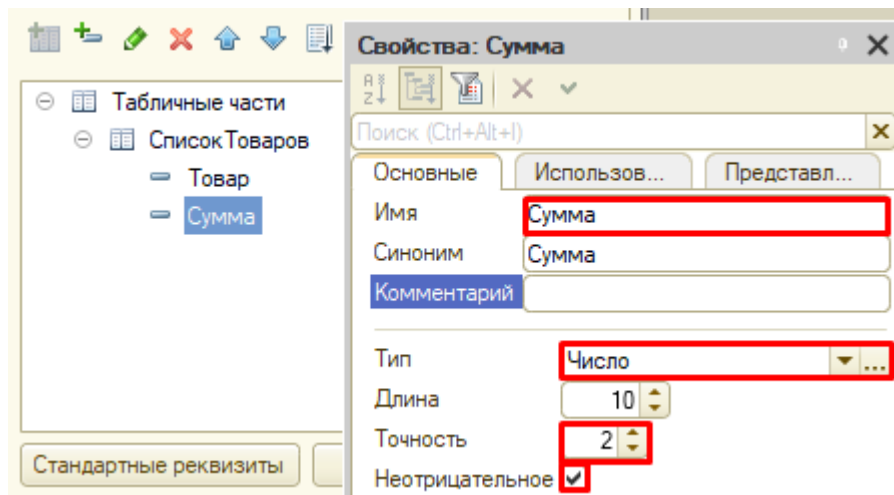
Поскольку в справочнике «Номенклатура» мы указывали для этого реквизита тип «Булево», то в значении параметра нужно указать именно его.



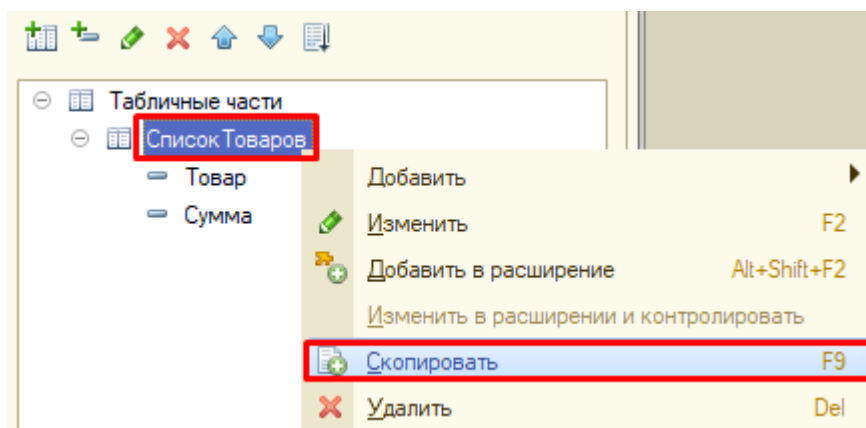
Для товаров значение «ЭтоУслуга» будет установлено в положении «Ложь».



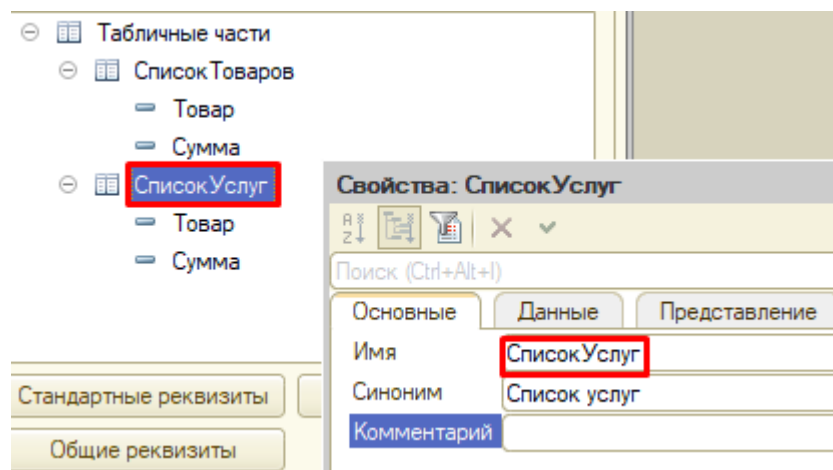
Далее добавим реквизит табличной части «Сумма».



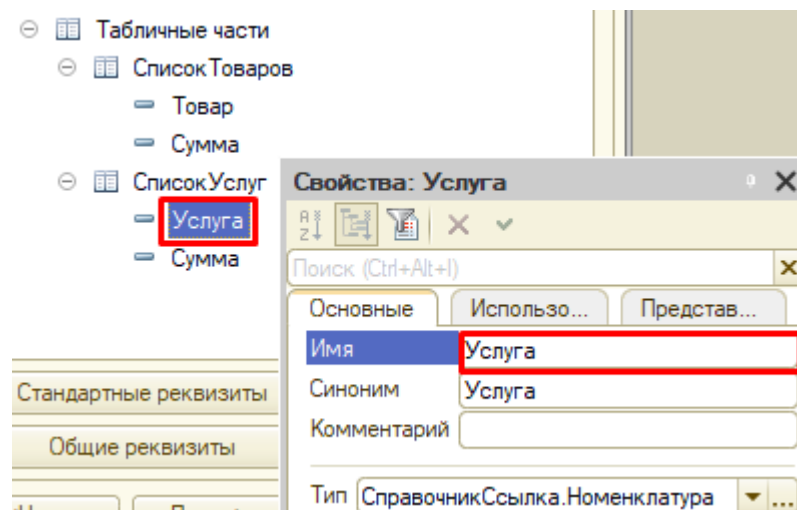
Табличная часть «СписокУслуг» структурно будет совпадать с уже созданной табличной частью «СписокТоваров». Для увеличения скорости разработки скопируем существующую табличную часть и несколько изменим ее.



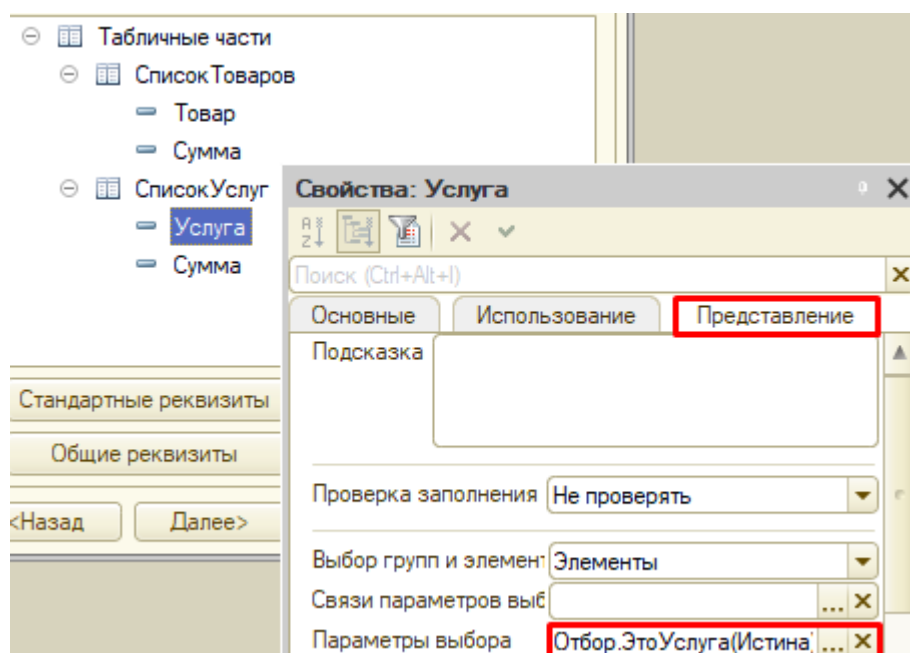
Во-первых, изменим свойство «Имя» у скопированной табличной части. Назовите табличную часть «СписокУслуг».



Во-вторых, изменим свойство «Имя» у реквизита табличной части «Товар». Поскольку данная табличная часть будет хранить перечень услуг, то и реквизиту нужно дать имя «Услуга».

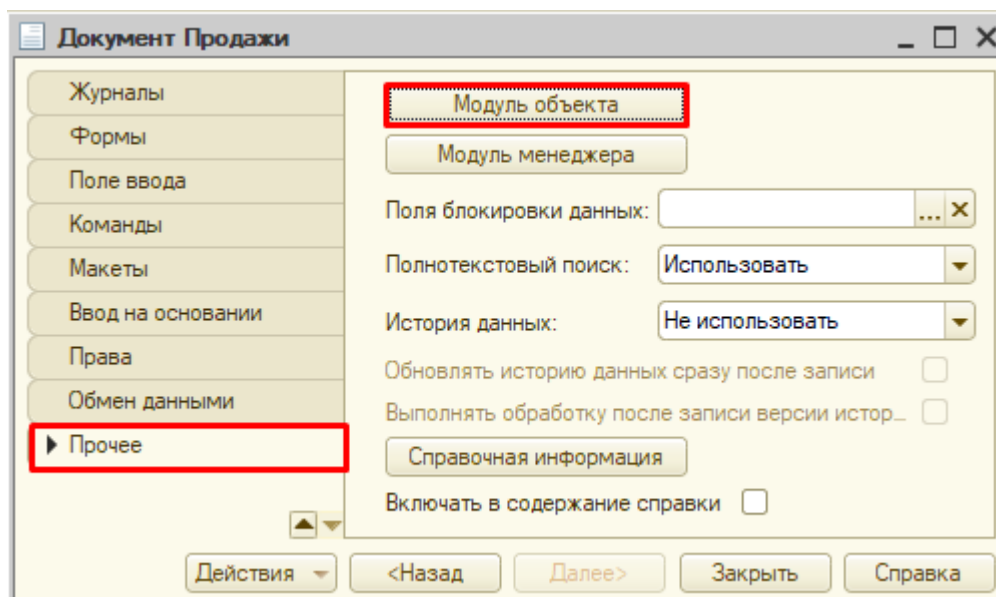


В-третьих, на вкладке «Представление» изменим параметры выбора: для услуг значение типа «Булево» будет в положении «Истина».



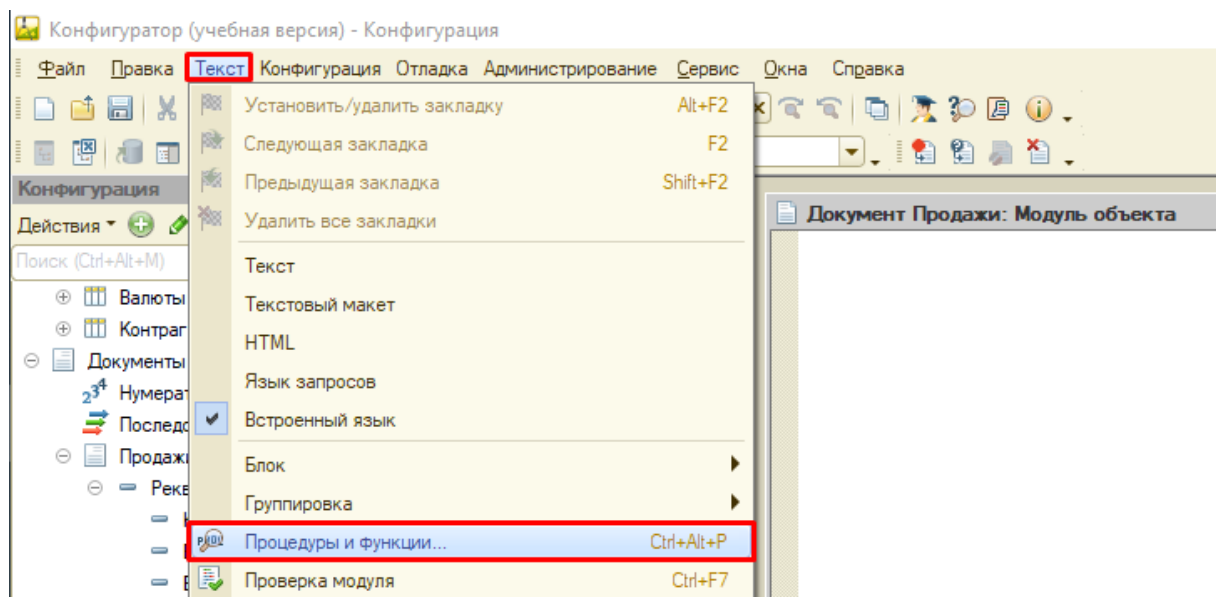
Чтобы итоговая стоимость формировалась автоматически, нам нужно описать событие в модуле документа «Продажи».

Для этого перейдем на вкладку «Прочее» и откроем «Модуль объекта».

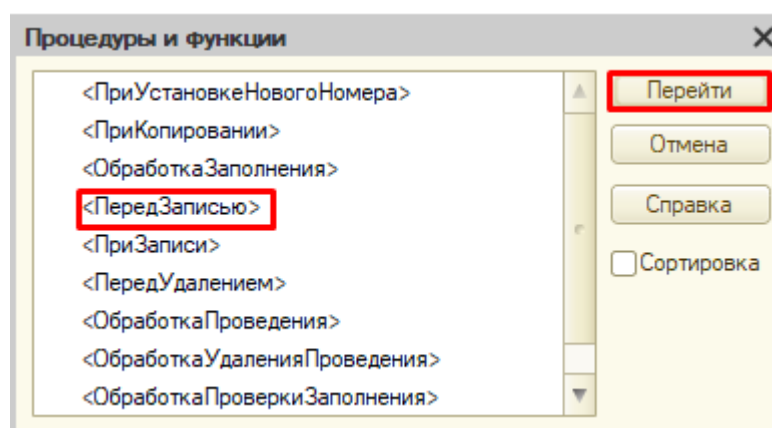


Перед нами откроется модуль объекта, в котором нужно определить событие, при наступлении которого будет происходить описываемый алгоритм.

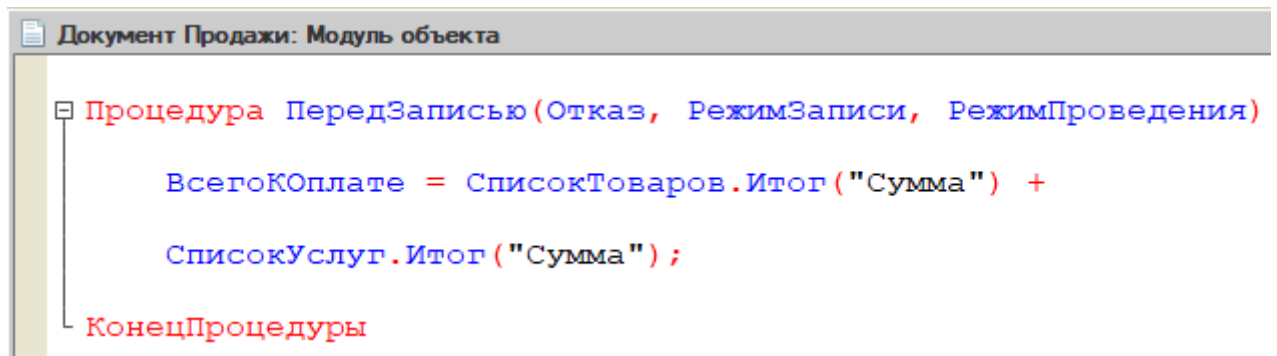
Нам понадобится системное событие, вызовем его через главное меню «Текст» ? «Процедуры и функции».



В открывшемся окне выберем «ПередЗаписью».

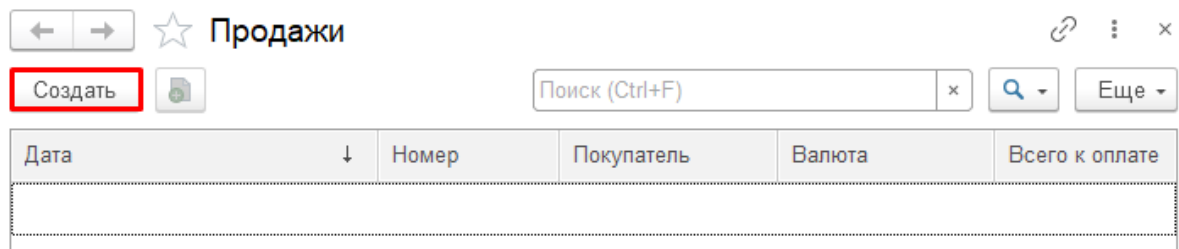


В модуле объекта сформируется обработчик события. Далее нам нужно описать подсчет итоговой суммы как итог по товарам и услугам.

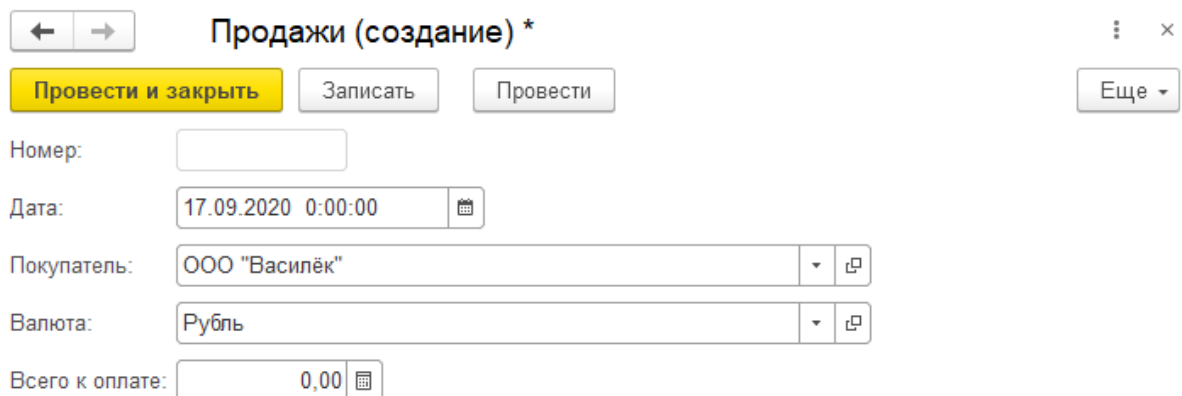


ВсегоКОплате = СписокТоваров.Итог("Сумма") + СписокУслуг.Итог("Сумма");

Запустим режим «1С:Предприятие» и создадим несколько документов.

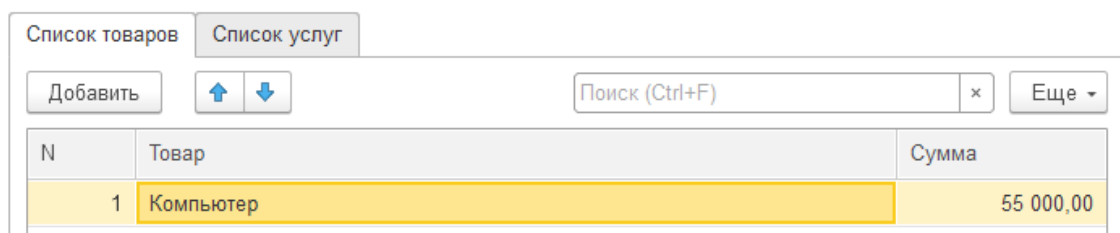


Заполним поля шапки документа, проигнорировав поле «Всего к оплате» (оно заполнится автоматически при записи).



Обратим внимание, что поле «Валюта» уже было заполнено по умолчанию.

Затем заполним таблицы «Список товаров» и «Список услуг».



Список товаров

Список услуг

Добавить

↑

↓

Поиск (Ctrl+F)

×

Еще ▾

N	Услуга	Сумма
1	Доставка	3 000,00

Внимание!

В список товаров можно добавлять только товары, а в список услуг – только услуги.

В момент записи или проведения документа поле «Всего к оплате» заполнится автоматически.

←

→

☆ Продажи 000000001 от 17.09.2020 14:19:39

🔗 ⋮ ×

Провести и закрыть

Записать

Провести

Еще ▾

Номер:

000000001

Дата:

17.09.2020 14:19:39

📅

Покупатель:

ООО "Василёк"

▾ 📄

Валюта:

Рубль

▾ 📄

Всего к оплате:

58 000,00

📄

Список товаров

Список услуг

Создайте еще несколько документов на разных покупателей и разные валюты.

←

→

☆ Продажи

🔗 ⋮ ×

Создать

📄

Поиск (Ctrl+F)

×

🔍 ▾

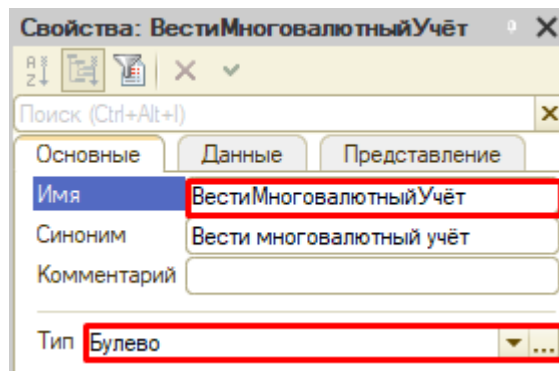
Еще ▾

Дата	↓	Номер	Покупатель	Валюта	Всего к оплате
📄 17.09.2020 14:20:49		000000001	ООО "Василёк"	Рубль	58 000,00
📄 17.09.2020 14:21:16		000000002	ООО "Василёк"	Доллар	1 445,00
📄 17.09.2020 14:21:49		000000003	ООО "Мак"	Рубль	130 000,00

Сейчас мы по умолчанию установили валюту «Рубль», оставив пользователю возможность выбора другой валюты. Но возможность использовать разные валюты нам нужно сделать опциональной. Для этого воспользуемся общим механизмом *функциональные опции*. Более подробно про функциональные опции можно прочитать здесь: <https://v8.1c.ru/platforma/funktsionalnaya-opciya/>.

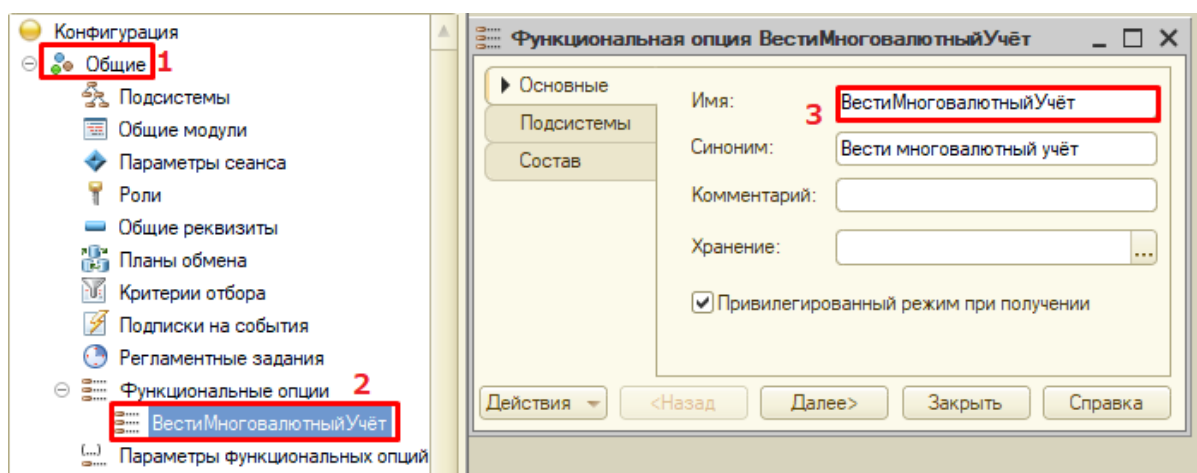
У пользователя должен быть физический способ включения и отключения многовалютного учета. Реализуем это с помощью обычной константы с типом «Булево».

Добавим константу «ВестиМноговалютныйУчет».



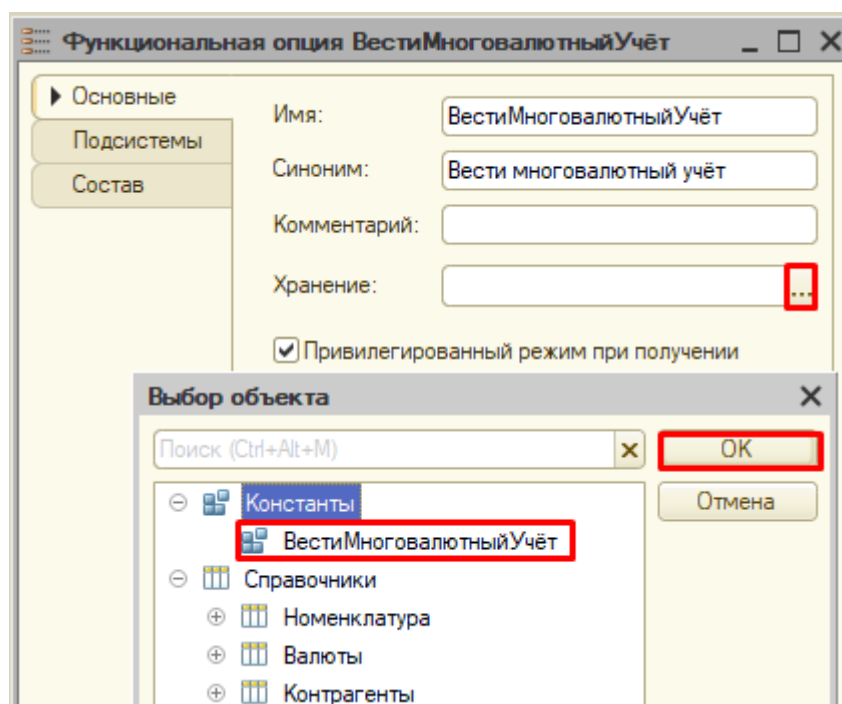
Теперь приступим к созданию и настройке функциональной опции.

В ветке «Общие» дерева конфигурации найдем раздел «Функциональные опции» и добавим опцию с именем «ВестиМноговалютныйУчет».



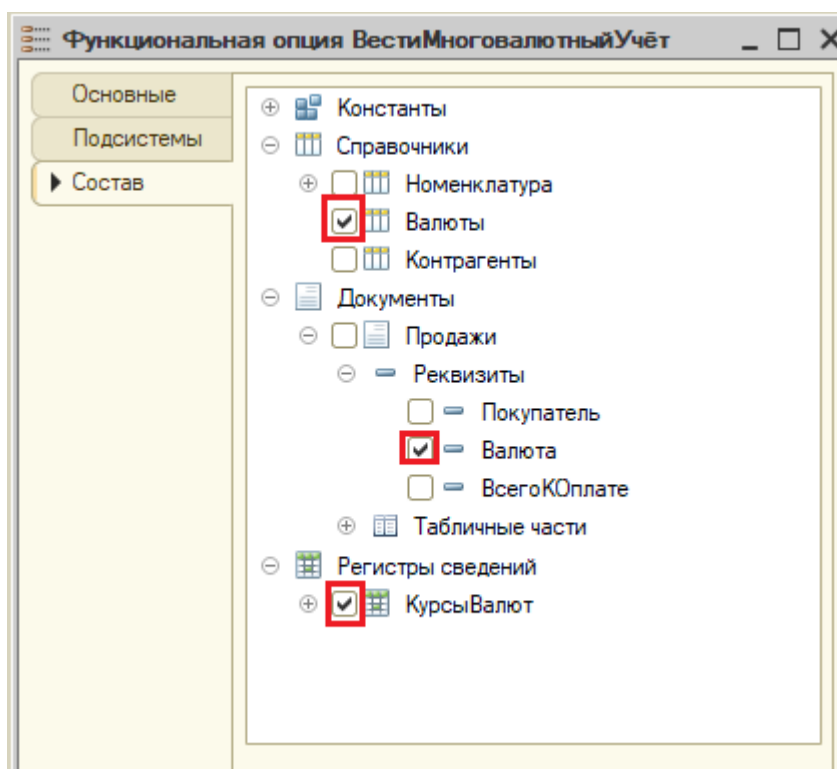
Теперь перейдем к ее настройке.

Первым делом нужно указать, где она будет храниться. Укажем в качестве места хранения заранее заготовленную константу с аналогичным именем.



Затем нужно указать, на что будет влиять наша функциональная опция. Она должна «включать» и «выключать» поле «Валюта» в документе «Продажи».

Укажем необходимые реквизиты на вкладке «Состав».



Запустим режим «1С:Предприятие» и проверим работоспособность.

← → ☆ Продажи

Создать [Иконка] Поиск (Ctrl+F) [Поиск] [Еще]

Дата	Номер	Покупатель	Валюта	Всего к оплате
17.09.2020 14:20:49	000000001	ООО "Василёк"	Рубль	58 000,00
17.09.2020 14:21:16	000000002	ООО "Василёк"	Доллар	1 445,00
17.09.2020 14:21:49	000000003	ООО "Мак"	Рубль	130 000,00

Открыв список документов, мы обнаружим, что отсутствует колонка «Валюта». Это связано с тем, что значение типа «Булево» по умолчанию – «Ложь». Чтобы вновь иметь возможность выбирать валюты, следует у константы перевести значение в положение «Истина» и перезапустить режим «1С:Предприятие».

При повторном открытии списка всех документов после проделанных действий мы обнаружим, что колонка «Валюта» снова отображается.

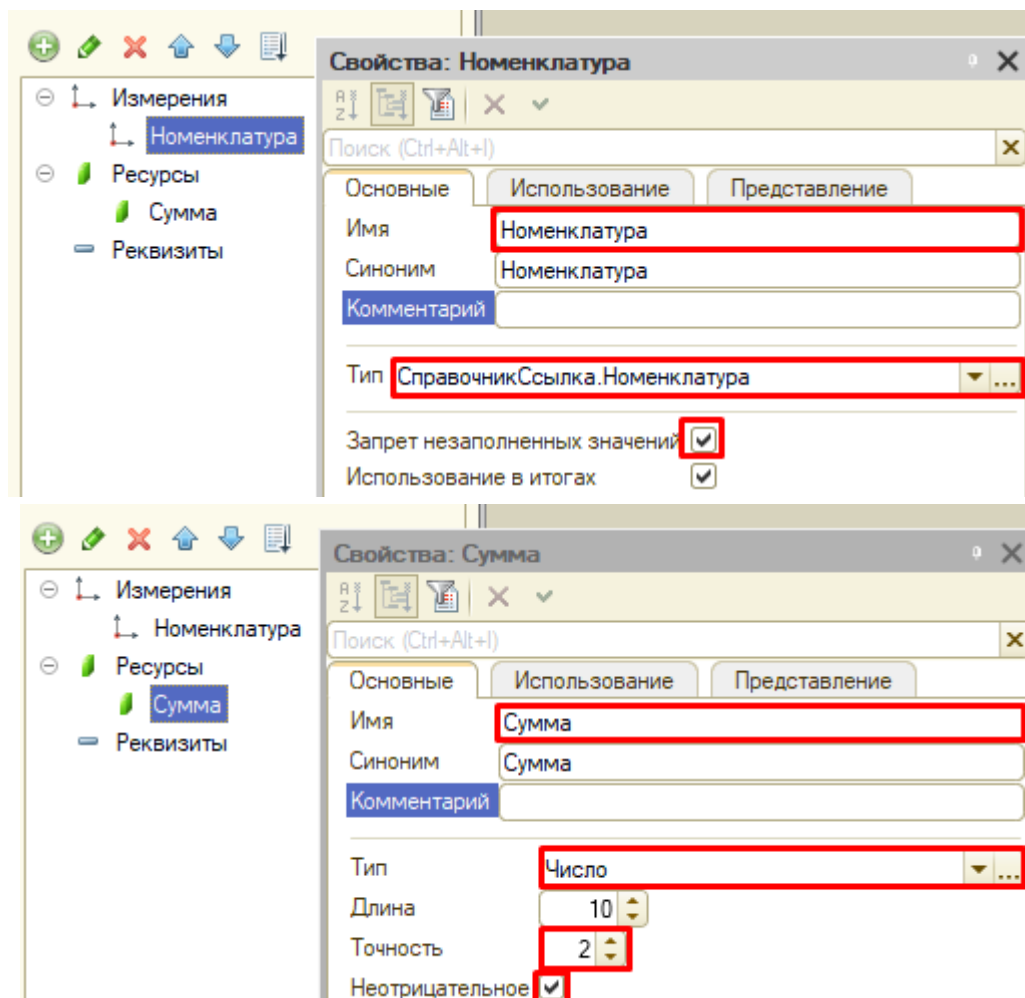
Дата	Номер	Покупатель	Валюта	Всего к оплате
17.09.2020 14:20:49	000000001	ООО "Василёк"	Рубль	58 000,00
17.09.2020 14:21:16	000000002	ООО "Василёк"	Доллар	1 445,00
17.09.2020 14:21:49	000000003	ООО "Мак"	Рубль	130 000,00

«В системе нужно реализовать хранилище суммы доходов в рублях по номенклатурным позициям».

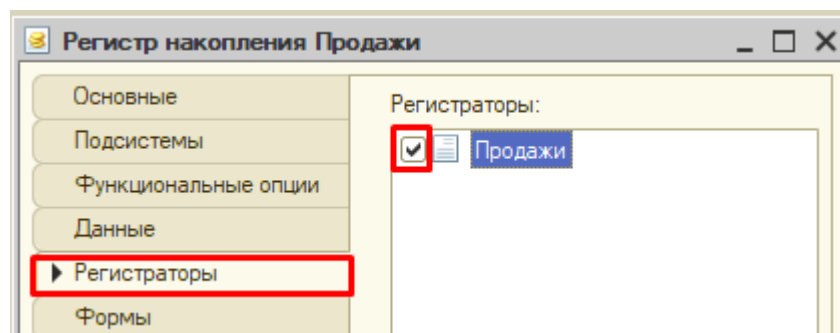
Для решения такого рода задачи нам потребуется *регистр накопления* вида «Обороты» (подробнее про регистры накоплений можно прочитать здесь: <https://v8.1c.ru/platforma/registr-nakopleniya/>).

Добавим *регистр накопления* «Продажи».

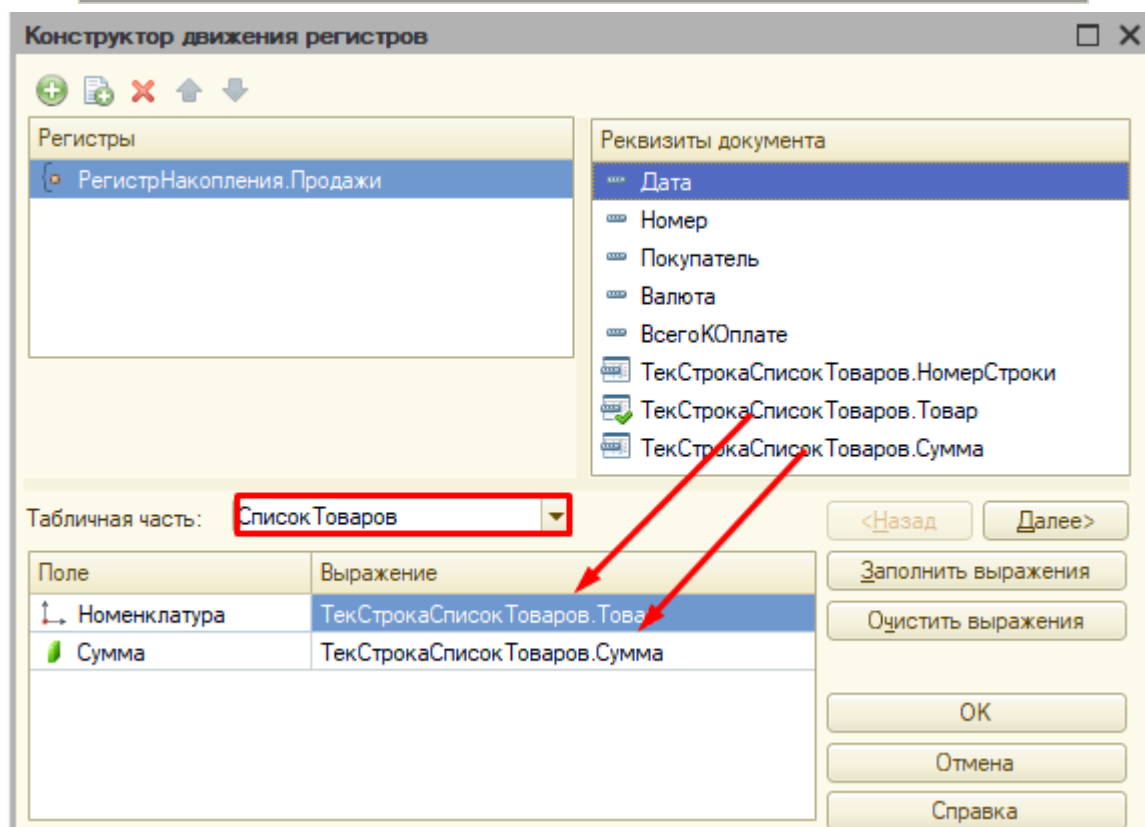
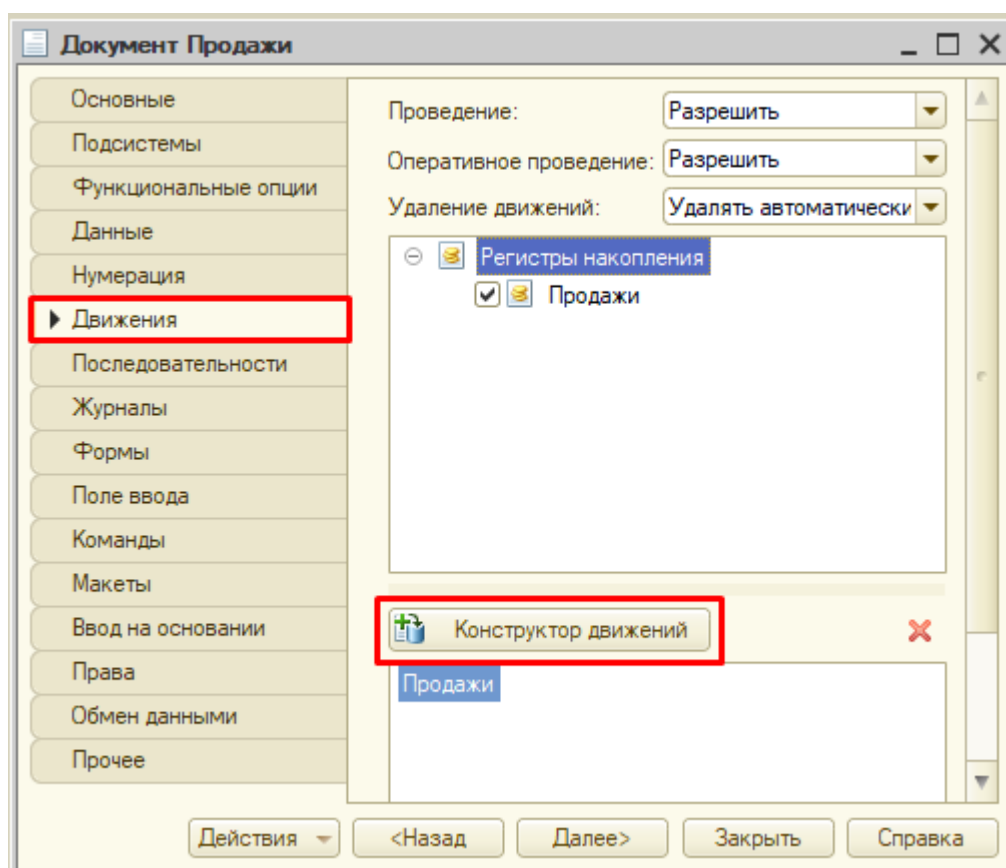
Как и в случае с *регистром сведений*, на вкладке «Данные» определим структуру нашего регистра. Отвечая на вопросы: «что?» и «в разрезе чего?», укажем, что в качестве измерения будет выступать «Номенклатура», а в качестве ресурса – «Сумма».



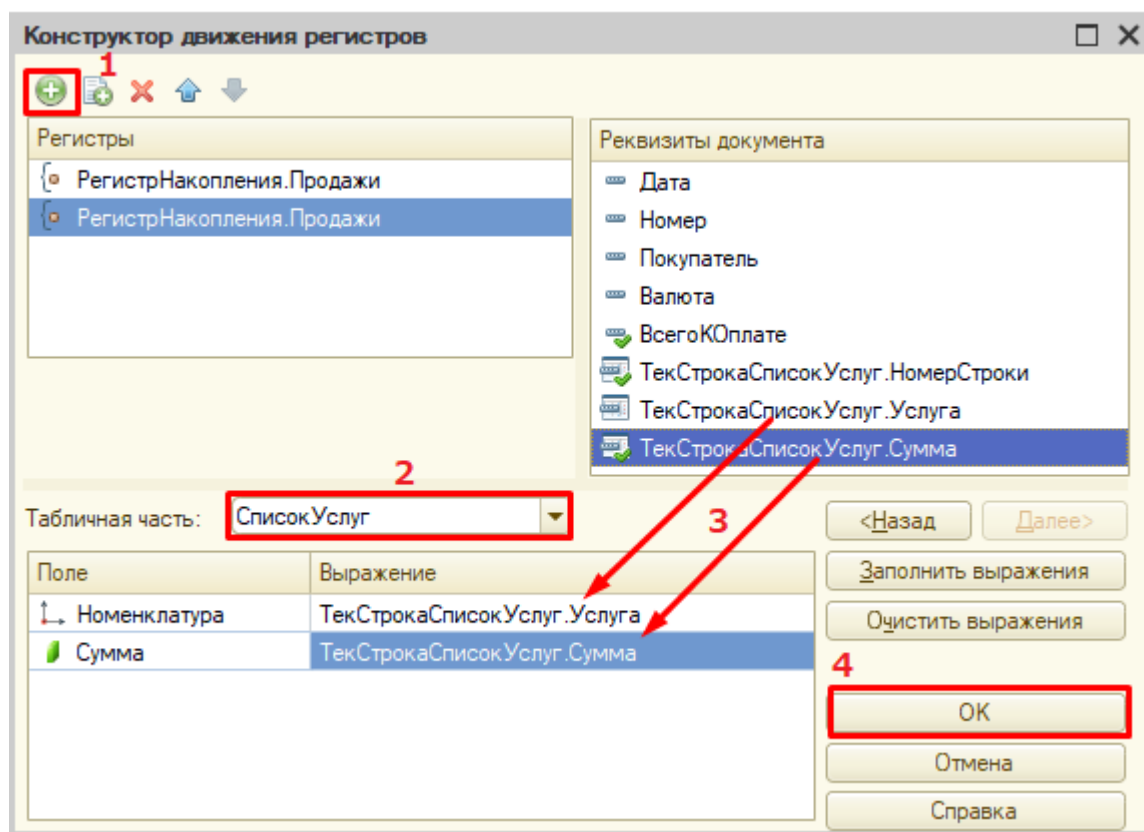
Далее на вкладке «Регистраторы» следует указать документ-регистратор для нашего регистра.



Но для совершения движений данных в регистр этого недостаточно. Необходимо описать правила с помощью *конструктора движений* документа «Продажи».



Но кроме табличной части с товарами есть и вторая часть – с услугами. Чтобы учитывать и ее – создадим еще одно движение в тот же регистр.



В модуле объекта (там же, где мы описывали итоговый подсчет) сформируется программный код, он будет совершать движения исключительно по числовым значениям реквизита «ВсегоКОплате», тем самым складывая рубли и другие валюты без разбора. Нам нужно это исправить.

Суммовые движения будут происходить как произведение суммы товара на его курс валюты, который мы будем получать с помощью функции – ее мы создадим позже.

```

Процедура ОбработкаПроведения (Отказ, Режим)

    Движения.Продажи.Записывать = Истина;
    Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
        Движение = Движения.Продажи.Добавить ();
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаСписокТоваров.Товар;
        Движение.Сумма = ТекСтрокаСписокТоваров.Сумма * ПолучитьКурсВалюты (Валюта, Дата);
    КонечЦикла;

    Движения.Продажи.Записывать = Истина;
    Для Каждого ТекСтрокаСписокУслуг Из СписокУслуг Цикл
        Движение = Движения.Продажи.Добавить ();
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаСписокУслуг.Услуга;
        Движение.Сумма = ТекСтрокаСписокУслуг.Сумма * ПолучитьКурсВалюты (Валюта, Дата);
    КонечЦикла;

КонечПроцедуры

* ПолучитьКурсВалюты(Валюта, Дата)

```

Следующим этапом будет написание функции, которая будет обращаться к *реестру сведений* и получать актуальный курс валют.

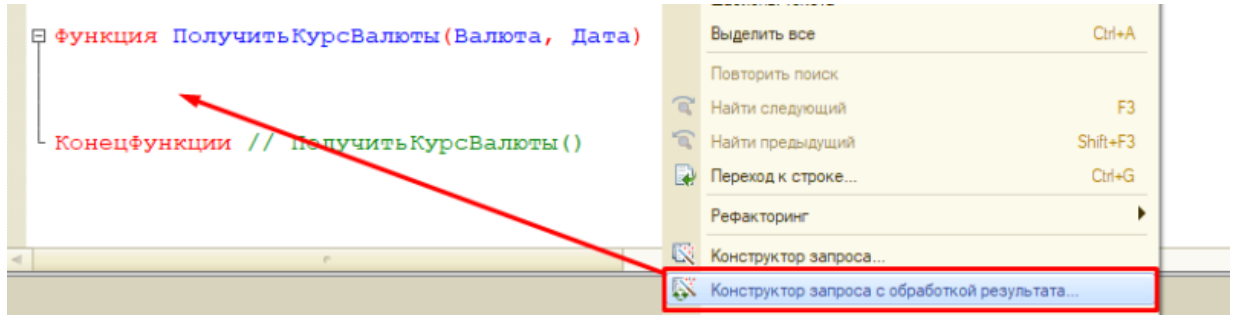
```

Функция ПолучитьКурсВалюты(Валюта, Дата)
...
КонецФункции // ПолучитьКурсВалюты()

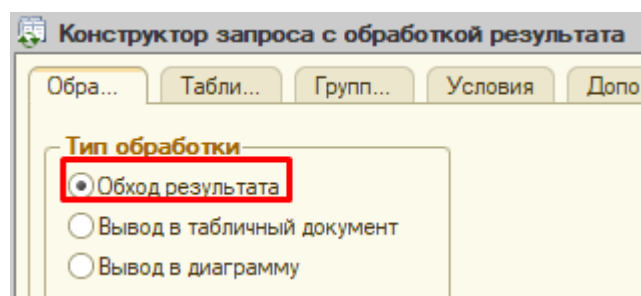
```

Информацию мы будем получать с помощью запроса к виртуальной таблице – нам нужен срез последних значений по указанным валюте и дате.

Установим курсор внутри функции и вызовем контекстное меню правой кнопкой мыши.



В открывшемся окне на вкладке «Обработка результата» следует выбрать тип обработки «Обход результата».



Переходим на вкладку «Таблицы и поля».

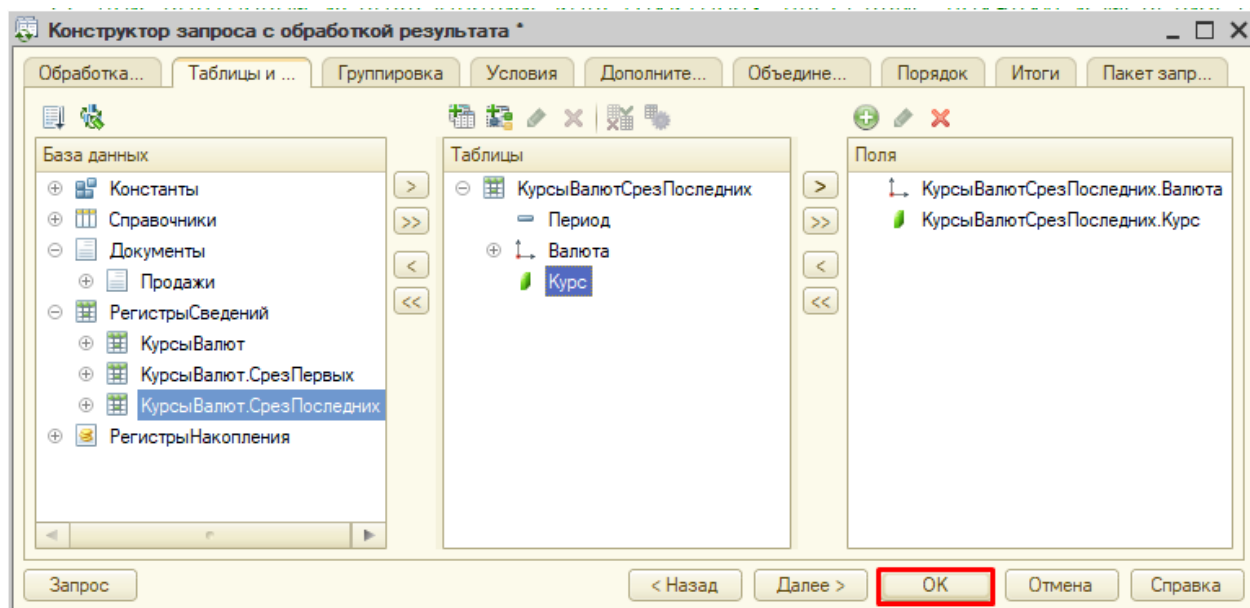
Открывается *конструктор запроса*. Эта вкладка имеет три части:

- Часть слева отображает все объекты конфигурации, имеющиеся в нашей базе данных. Нужно выбрать лишь те объекты, из которых мы хотим получать данные.
- Посередине находятся таблицы – это выбранные нами объекты, откуда мы хотим получать данные для конкретного отчета.
- Справа поля – это те значения (поля), которые мы хотим увидеть в отчете.

Данные будем брать не из *регистра сведений* напрямую, а из виртуальной таблицы, которую создает этот регистр автоматически. Данная виртуальная таблица позволит получить последние введенные значения по каждой валюте.

Чтобы перенести данные из одного окна в другое, просто перетаскивайте нужные поля с помощью мыши либо воспользуйтесь стрелочками, расположенными между окнами.

В результате данное окно должно быть заполнено следующим образом:



Далее нужно преобразовать полученный программный код.

```

Функция ПолучитьКурсВалюты(Валюта, Дата)

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |   КурсыВалютСрезПоследних.Валюта КАК Валюта,
        |   КурсыВалютСрезПоследних.Курс КАК Курс
        |ИЗ
        |   РегистрСведений.КурсыВалют.СрезПоследних КАК КурсыВалютСрезПоследних";

    РезультатЗапроса = Запрос.Выполнить();

    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

    ВыборкаДетальныеЗаписи.Следующий();

КонецФункции // ПолучитьКурсВалюты()
  
```

Виртуальная таблица позволяет использовать встроенные методы для работы с параметрами. Такими параметрами у нас будут выступать *период* и *валюта*.

```

Запрос.Текст =
    "ВЫБРАТЬ
    |   КурсыВалютСрезПоследних.Валюта КАК Валюта,
    |   КурсыВалютСрезПоследних.Курс КАК Курс
    |ИЗ
    |   РегистрСведений.КурсыВалют.СрезПоследних(&Период, Валюта = &Валюта) КАК КурсыВалютСрезПоследних";
  
```

Далее нужно установить значения этих параметров. Укажем их как передаваемые в функцию параметры «Валюта» и «Дата».

```

    |   РегистрСведений.КурсыВалют.СрезПоследних(&Период, Валюта = &Валюта) КАК КурсыВалютСрезПоследних";

    Запрос.УстановитьПараметр("Валюта", Валюта);
    Запрос.УстановитьПараметр("Период", НачалоДня(Дата));

    РезультатЗапроса = Запрос.Выполнить();
  
```

Поскольку мы создаем функцию, то в самом конце нам нужно вернуть значение курса валюты.

ВыборкаДетальныеЗаписи.Следующий ();

Возврат ВыборкаДетальныеЗаписи.Курс;

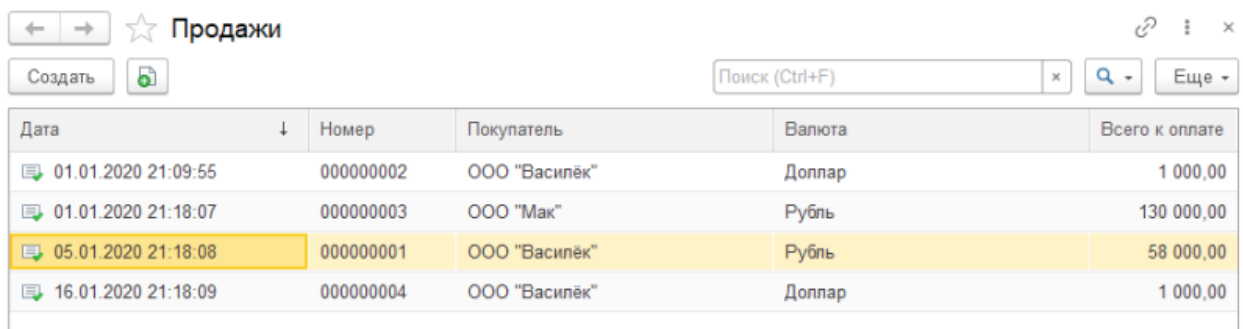
Конецфункции // ПолучитьКурсВалюты()

Итоговый программный код функции «ПолучитьКурсВалюты» будет таким:

```
Функция ПолучитьКурсВалюты(Валюта, Дата)
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     КурсыВалютСрезПоследних.Валюта КАК Валюта,
        |     КурсыВалютСрезПоследних.Курс КАК Курс
        | ИЗ
        |     РегистрСведений.КурсыВалют.СрезПоследних (&Период, Валюта = &Валюта) КАК КурсыВалютСрезПоследних";
    Запрос.УстановитьПараметр("Валюта", Валюта);
    Запрос.УстановитьПараметр("Период", НачалоДня(Дата));
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    ВыборкаДетальныеЗаписи.Следующий();
    Возврат ВыборкаДетальныеЗаписи.Курс;
Конецфункции // ПолучитьКурсВалюты()
```

Чтобы движения были совершены в *регистр накопления*, необходимо перепровести (провести повторно) созданные документы.

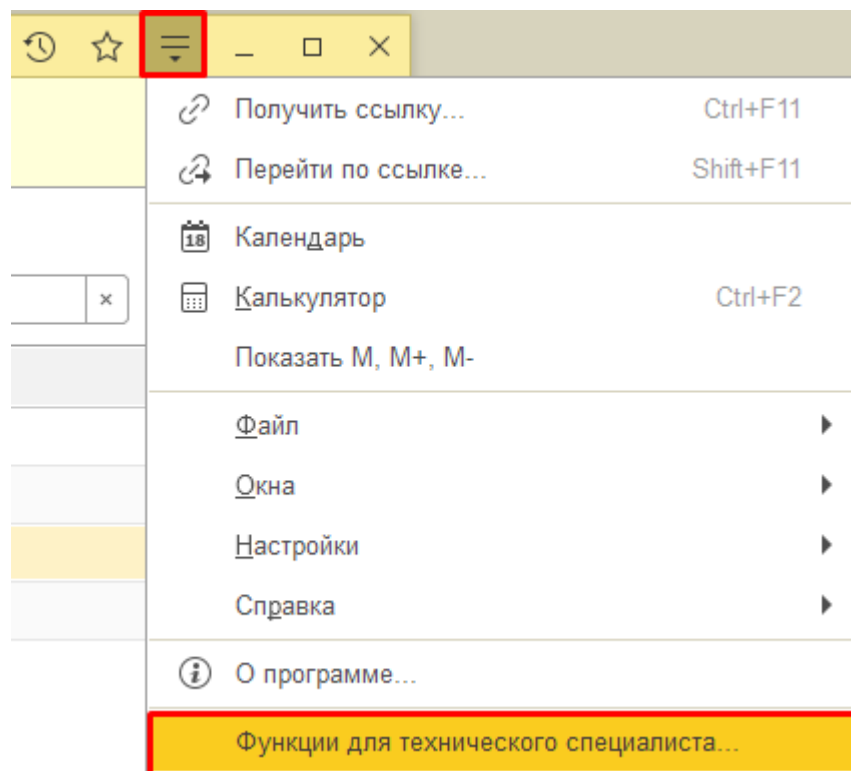
Запустим режим «1С:Предприятие» и сделаем это.



Дата	Номер	Покупатель	Валюта	Всего к оплате
01.01.2020 21:09:55	000000002	ООО "Василёк"	Доллар	1 000,00
01.01.2020 21:18:07	000000003	ООО "Мак"	Рубль	130 000,00
05.01.2020 21:18:08	000000001	ООО "Василёк"	Рубль	58 000,00
16.01.2020 21:18:09	000000004	ООО "Василёк"	Доллар	1 000,00

Обратите внимание, что на главной странице система не создала кнопку открытия регистра накопления. Это связано с тем, что все расчеты в регистрах накопления происходят в фоновом режиме, «за кадром», то есть пользователю о них знать не нужно вовсе. Поэтому по умолчанию регистры накопления настраивают так, чтобы пользователи не имели к ним доступа.

Но мы, будучи разработчиками, можем обратиться к любому объекту конфигурации. Для этого воспользуемся функциями для технического специалиста.



В общем перечне элементов следует найти *регистр накопления* «Продажи» и открыть его.

+ Регистры сведений
 - Регистры накопления
 Продажи
 T Регистры бухгалтерии

← → ☆ Продажи

Поиск (Ctrl+F) [x] [Q] [Еще ▾]

Период	Регистратор	Номер строки	Номенклатура	Сумма
• 01.01.2020 21:09:55	Продажи 000000002 от 01.01.202...	1	Компьютер	48 000,00
• 01.01.2020 21:09:55	Продажи 000000002 от 01.01.202...	2	Доставка	12 000,00
• 01.01.2020 21:18:07	Продажи 000000003 от 01.01.202...	1	Компьютер	80 000,00
• 01.01.2020 21:18:07	Продажи 000000003 от 01.01.202...	2	Телефон	45 000,00
• 01.01.2020 21:18:07	Продажи 000000003 от 01.01.202...	3	Доставка	5 000,00
• 05.01.2020 21:18:08	Продажи 000000001 от 05.01.202...	1	Компьютер	55 000,00
• 05.01.2020 21:18:08	Продажи 000000001 от 05.01.202...	2	Доставка	3 000,00
• 16.01.2020 21:18:09	Продажи 000000004 от 16.01.202...	1	Компьютер	56 000,00
• 16.01.2020 21:18:09	Продажи 000000004 от 16.01.202...	2	Доставка	14 000,00

[Σ] [↑] [↓] [↔]

Если соотнести результаты, например, по движениям за 16 января, то заметим, что в регистр пришли значения с учетом указанного за 15 января курса, так как 16 числа не было изменений в курсе.

«Необходимо построить «Отчет по продажам» с упорядочиванием по сумме доходов».

Построим отчет. Для этого воспользуемся соответствующим объектом конфигурации.

Определение

Отчет – это такой объект конфигурации, который формирует данные в понятном для пользователя виде, например, в виде таблицы или диаграммы (подробнее про отчеты можно прочитать здесь: <https://v8.1c.ru/platforma/otchet/>).

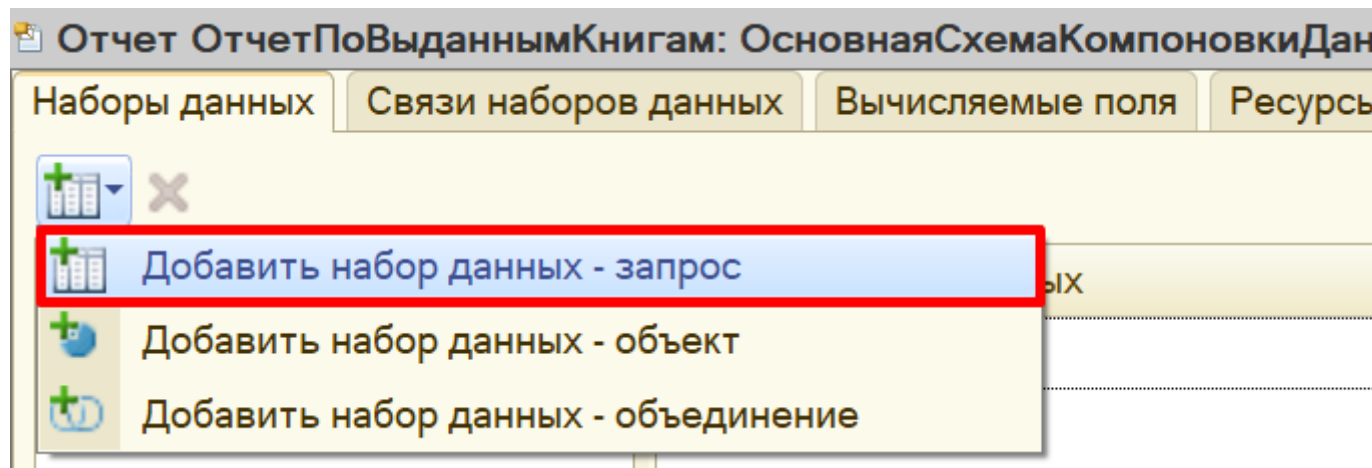
Добавим отчет «Продажи». Воспользуемся схемой компоновки данных.

The image shows two overlapping windows from a software application.

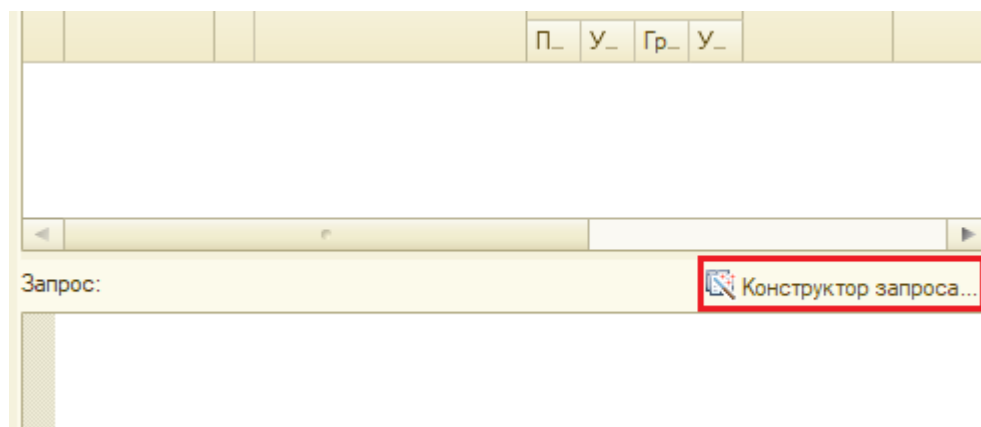
The top window, titled "Отчет Продажи", has a sidebar on the left with a tree view containing: "Основные" (selected), "Подсистемы", "Функциональные опции", "Данные", "Формы", "Команды", "Макеты", "Права", and "Прочее". The main area contains fields for "Имя:" (filled with "Продажи"), "Синоним:" (filled with "Продажи"), and "Комментарий:". Below these is a field for "Основная схема компоновки данных:" with a search icon and a button labeled "Открыть схему компоновки данных" which is highlighted with a red rectangle.

The bottom window, titled "Конструктор макета", contains fields for "Имя:" (filled with "ОсновнаяСхемаКомпоновкиДанных"), "Синоним:" (filled with "Основная схема компоновки данных"), and "Комментарий:". Below these is a section "Выберите тип макета:" with a list of radio button options: "Табличный документ", "Текстовый документ", "Двоичные данные", "Active document", "HTML документ", "Географическая схема", "Графическая схема", "Схема компоновки данных" (which is selected), "Макет оформления компоновки данных", and "Внешняя компонента". At the bottom is a field "Загрузить из файла:" with a search icon. At the very bottom are three buttons: "Готово" (highlighted with a red rectangle), "Отмена", and "Справка".

Добавим новый запрос к базе данных.



Для формирования запроса воспользуемся *конструктором запроса*.



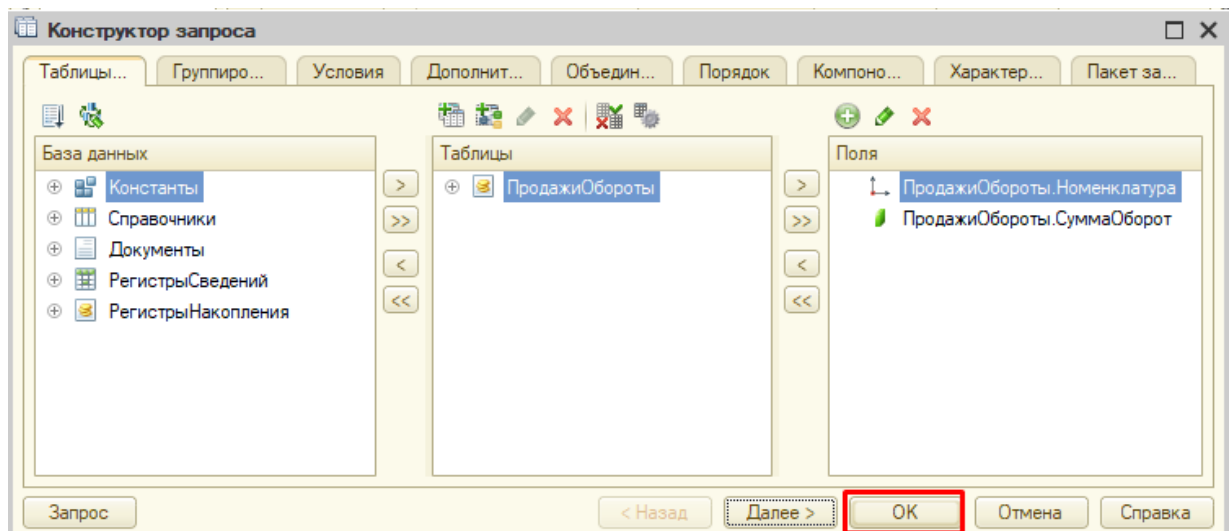
Открывается *конструктор запроса*. Эта вкладка имеет три части:

- Часть слева отображает все объекты конфигурации, имеющиеся в нашей базе данных. Нужно выбрать лишь те объекты, из которых мы хотим получать данные.
- Посередине находятся таблицы – это выбранные нами объекты, откуда мы хотим получать данные для конкретного отчета.
- Справа поля – это те значения (поля), которые мы хотим увидеть в отчете.

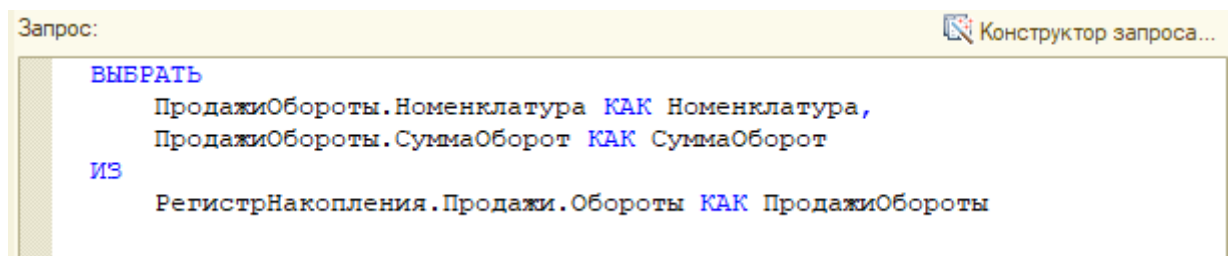
Данные будем брать из *регистра накопления* напрямую, чтобы иметь возможность рассчитывать средний балл.

Чтобы перенести данные из одного окна в другое, просто перетащите нужные поля с помощью мыши либо воспользуйтесь стрелочками, расположенными между окнами.

В результате данное окно должно быть заполнено следующим образом:



У нас сформируется запрос на встроенном языке запросов 1С.

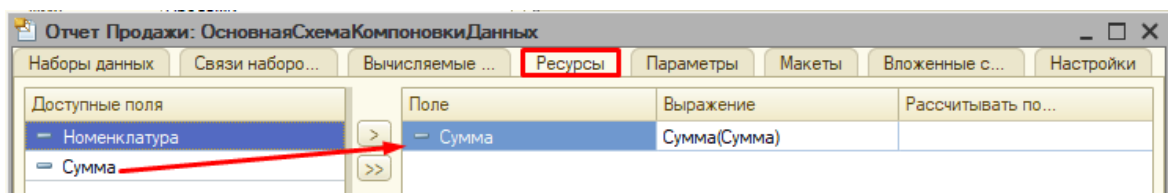


Для красоты зададим синоним полю «СуммаОборот». Сделать это можно с помощью конструктора запроса либо прямо в окне запроса.

```

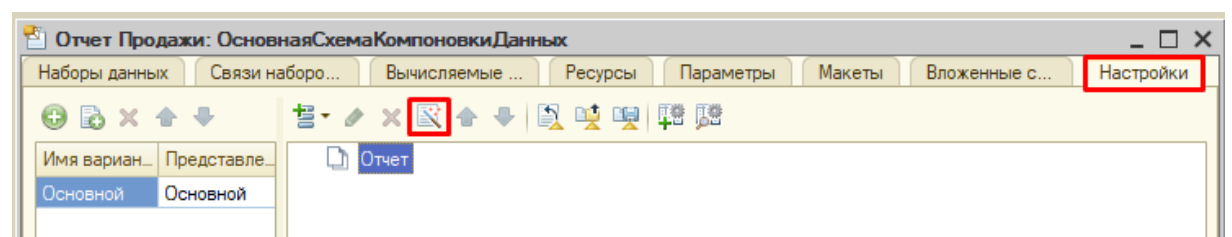
ВЫБРАТЬ
    ПродажиОбороты.Номенклатура КАК Номенклатура,
    ПродажиОбороты.СуммаОборот КАК Сумма
ИЗ
    РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты
  
```

Далее мы реализуем итоговый подсчет всех сумм для нашего отчета, указав поле «Сумма» в качестве ресурса на соответствующей вкладке.



Теперь переходим на вкладку «Настройки» для оформления внешнего вида отчета.


Воспользуемся конструктором настроек отчета.





Построим отчет в виде списка.

Конструктор настроек компоновки данных

Тип отчета:

 ☒ Список. Данные по всем измерениям отчета выводятся списком.

 ☐ Таблица. Данные выводятся по измерениям расположенным как по горизонтали, так и по вертикали.

 ☐ Диаграмма. Данные выводятся в виде диаграммы.

< Назад **Далее >** ОК Отмена Справка

Выберем поля, которые будут отображаться в отчете.

Конструктор настроек компоновки данных

Поле
Номенклатура
Склад
Сумма

Выберите поля, которые будут отображаться в отчете.

Доступные поля

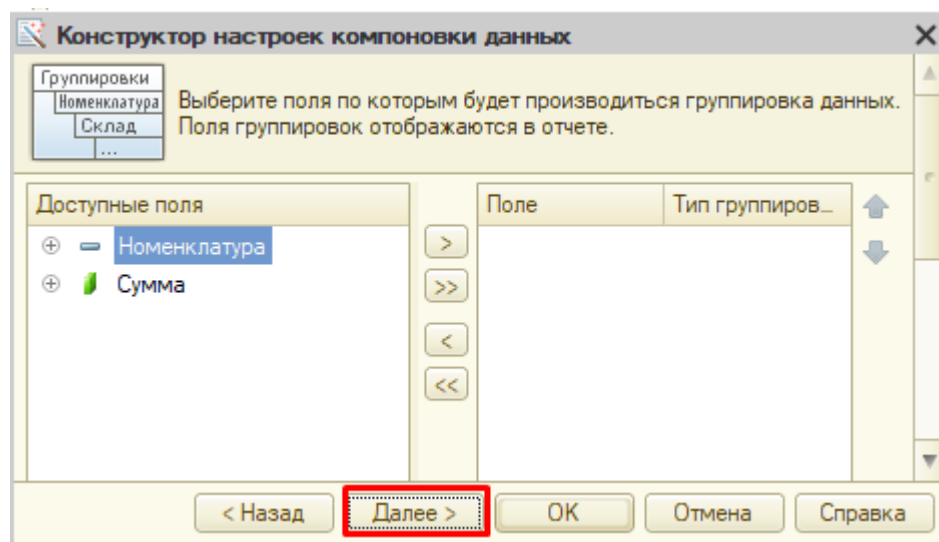
- Номенклатура
- Сумма
- СистемныеПоля
- ПараметрыДанных

Поле

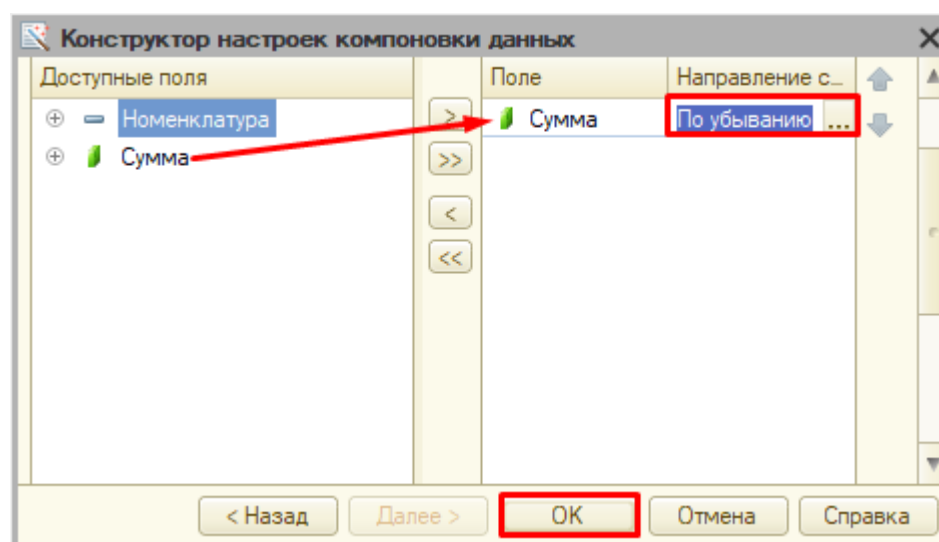
- Номенклатура
- Сумма

< Назад **Далее >** ОК Отмена Справка

Этап группировки пропустим.



Отсортируем отчет по возрастанию поля «Сумма».



Отчет готов. Запустим систему в режиме «1С:Предприятие» и построим отчет, указав разные валюты.

←

→

☆

Продажи

Сформировать

Выбрать вариант...

Настройки...

Номенклатура	Сумма
Компьютер	239 000,00
Телефон	45 000,00
Доставка	34 000,00
Итого	318 000,00

Поставленная задача решена.