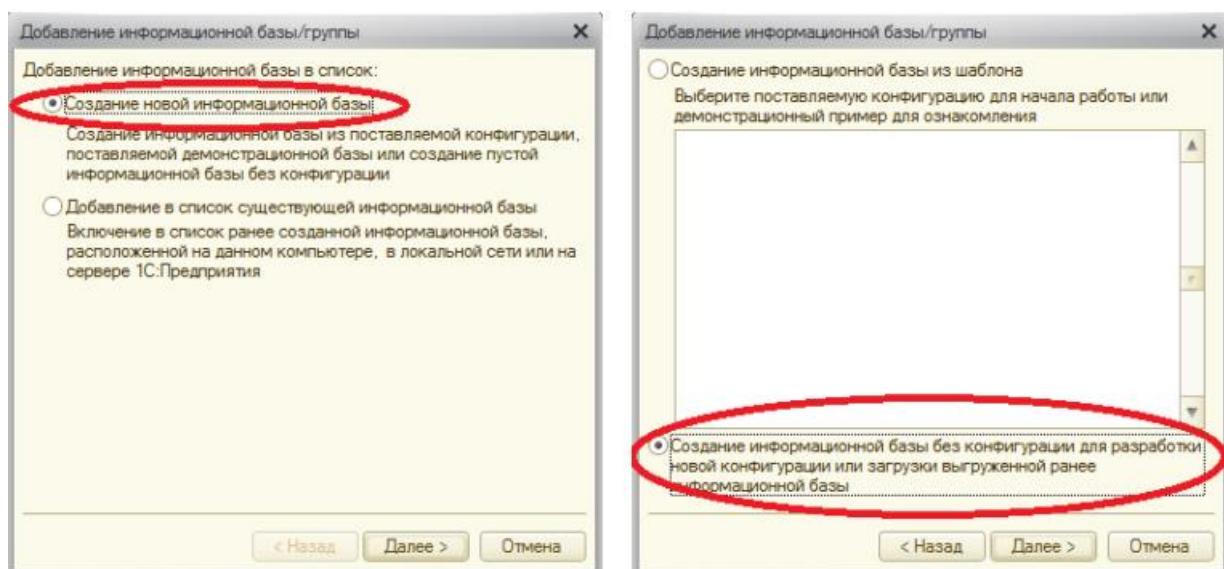
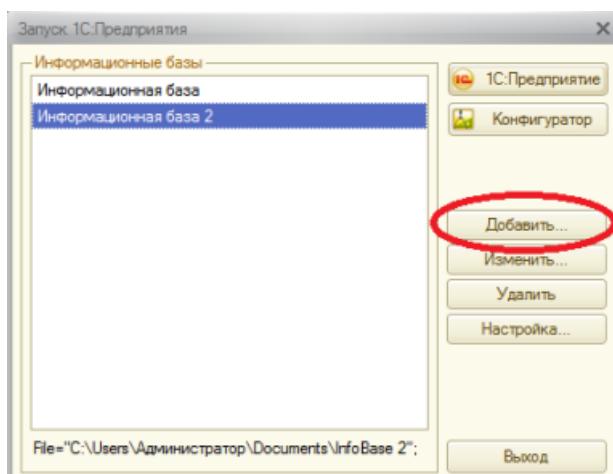


**Для выполнения задания Модуля 1:** Разработка модулей программного обеспечения для компьютерных систем нам необходимо разработать конфигурацию «Учета заявок на ремонт оборудования». Прежде чем приступить к работе ознакомьтесь с документацией, предоставленной в ресурсах.

### ***Создание новой информационной базы.***

Запустите 1С: Предприятие. В открывшемся диалоге Вы увидите список информационных баз, с которыми Вы работаете. Если список пуст, система предложит Вам создать новую базу. В любом случае, нажмите кнопку **Добавить...**.

В открывшемся диалоге выберите пункт **Создание новой информационной базы**. Нажмите **далее**, выберите пункт **Создание информационной базы** без конфигурации...

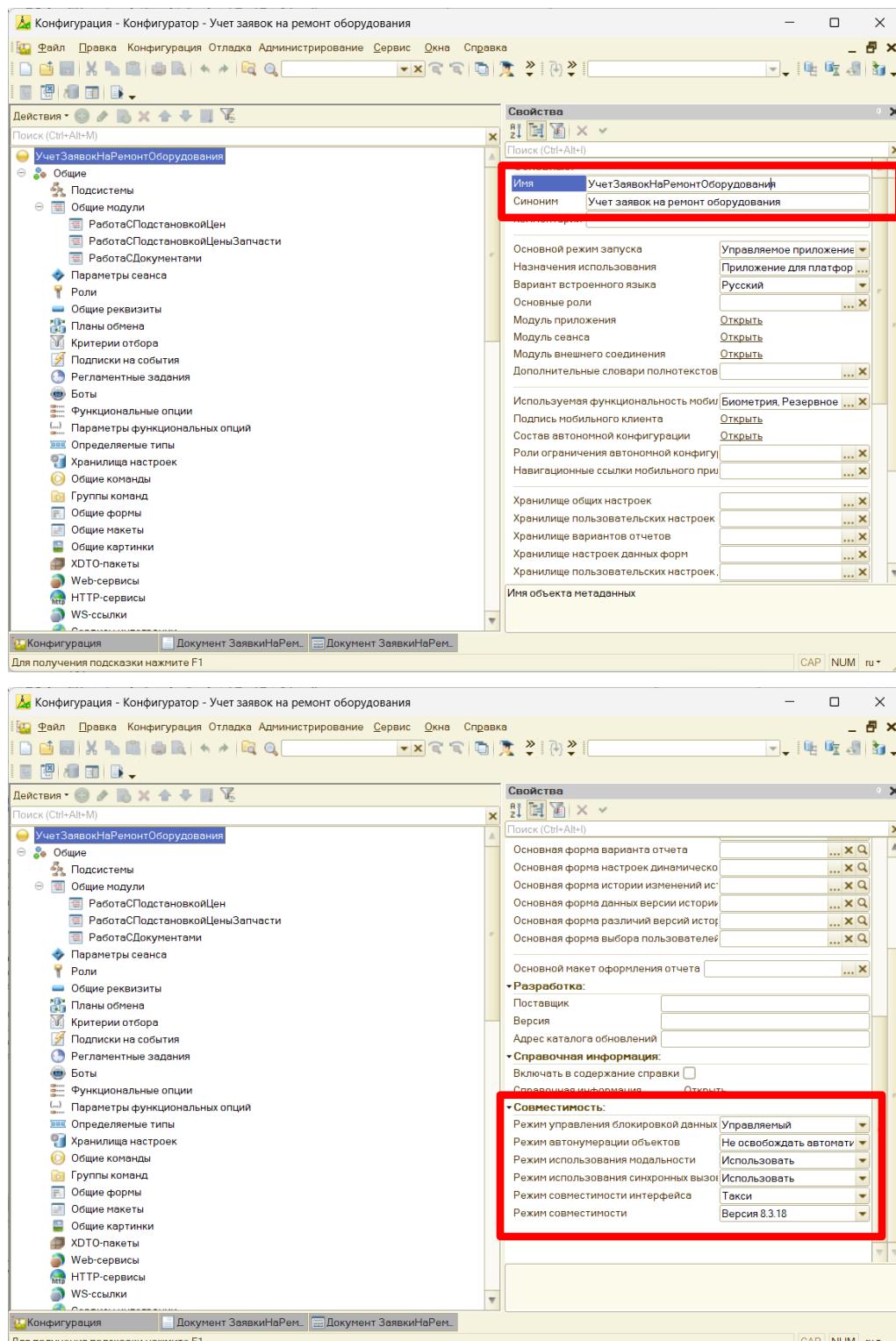


Нажмите кнопку **Далее**. Задайте наименование Вашей информационной базы согласно предметной области «Разработка программного модуля для учета заявок на ремонт оборудования» и выберите тип ее расположения на данном компьютере, где будут храниться результаты Вашей работы.

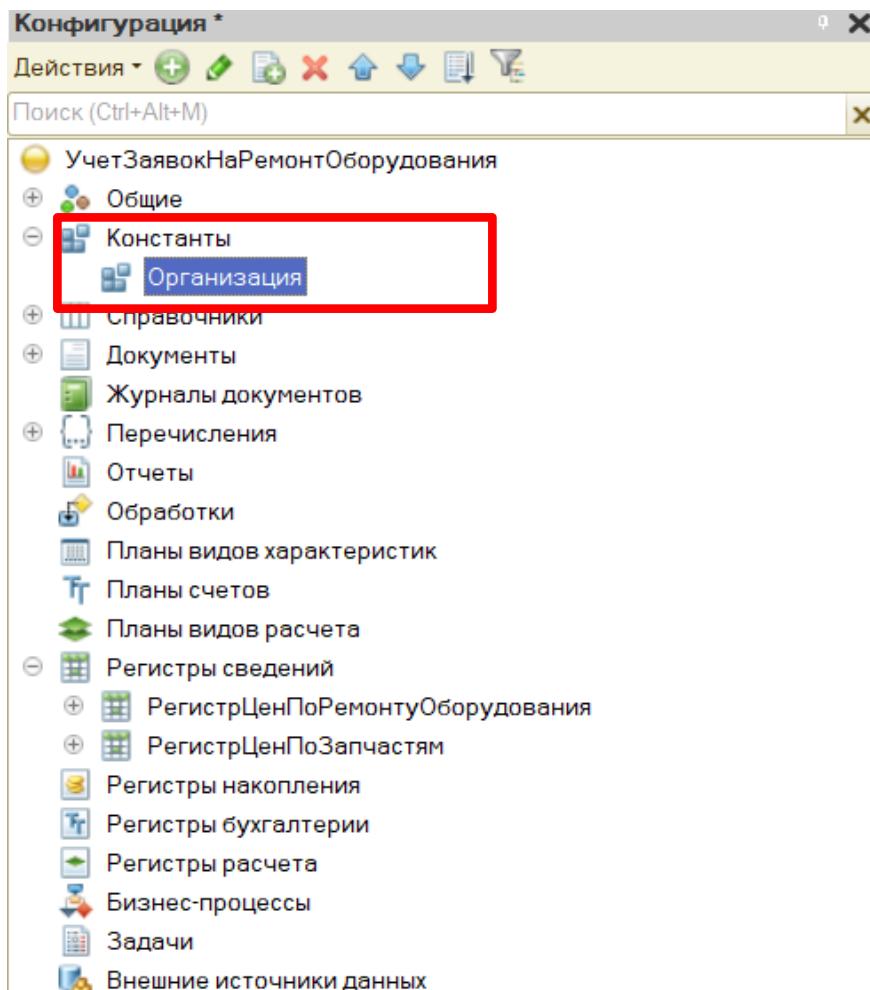
В диалоге запуска 1С: Предприятия, в списке информационных баз, Вы увидите созданную вами новую пустую базу.

Запустите 1С: Предприятие в режиме **Конфигуратора**.

В окне свойств задайте имя Вашей конфигурации



Создадим сначала константу, в котором будут храниться наименование организации. Тип данных *строка*, а длину *наименования* установите в 25 символов.

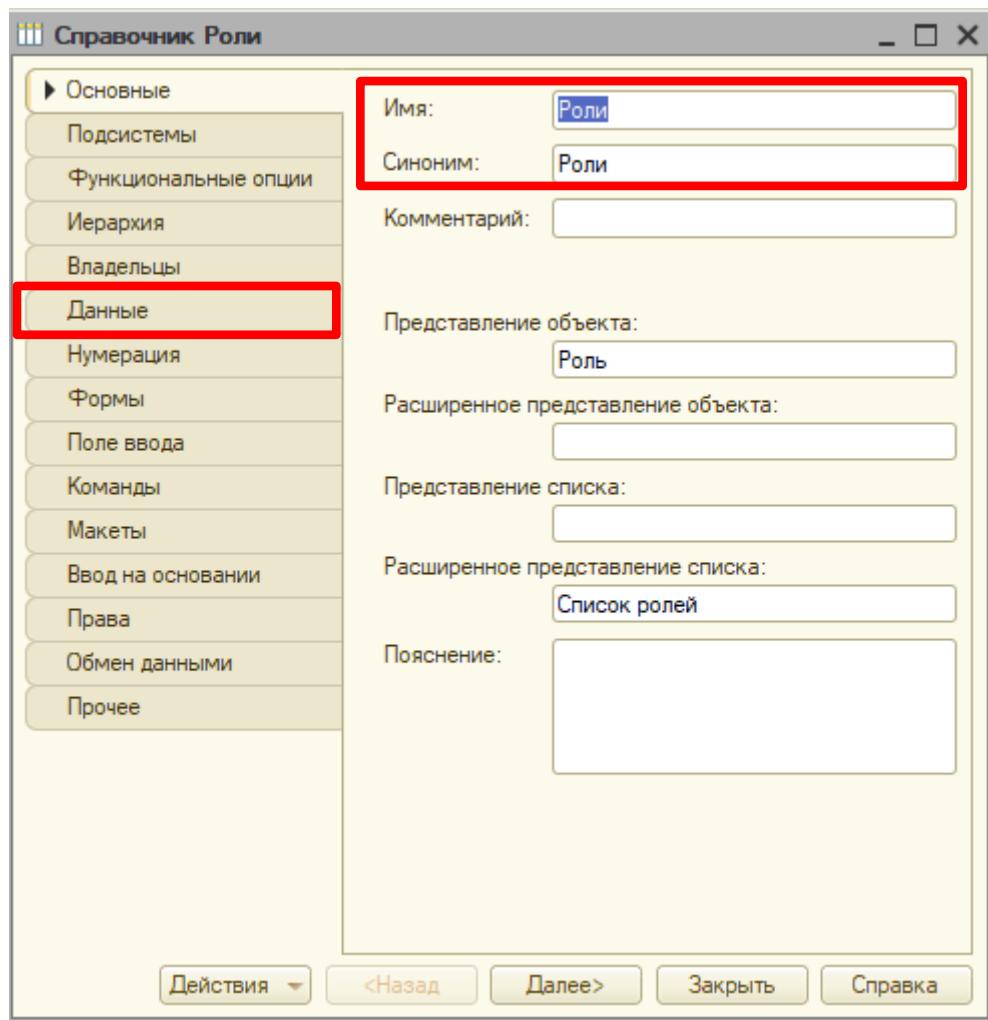


В режиме пользователя в Константу запишите название организации и сохраните.

Далее создадим такие объекты конфигурации как справочники. Создадим сначала справочник, в котором будут храниться данные о роль, которые будет выполнять пользователь.

Откройте в конфигураторе нашу конфигурацию, выделите в дереве объектов ветку **Справочники**, нажмите *Добавить*. Назовите **Справочник – Роль**.

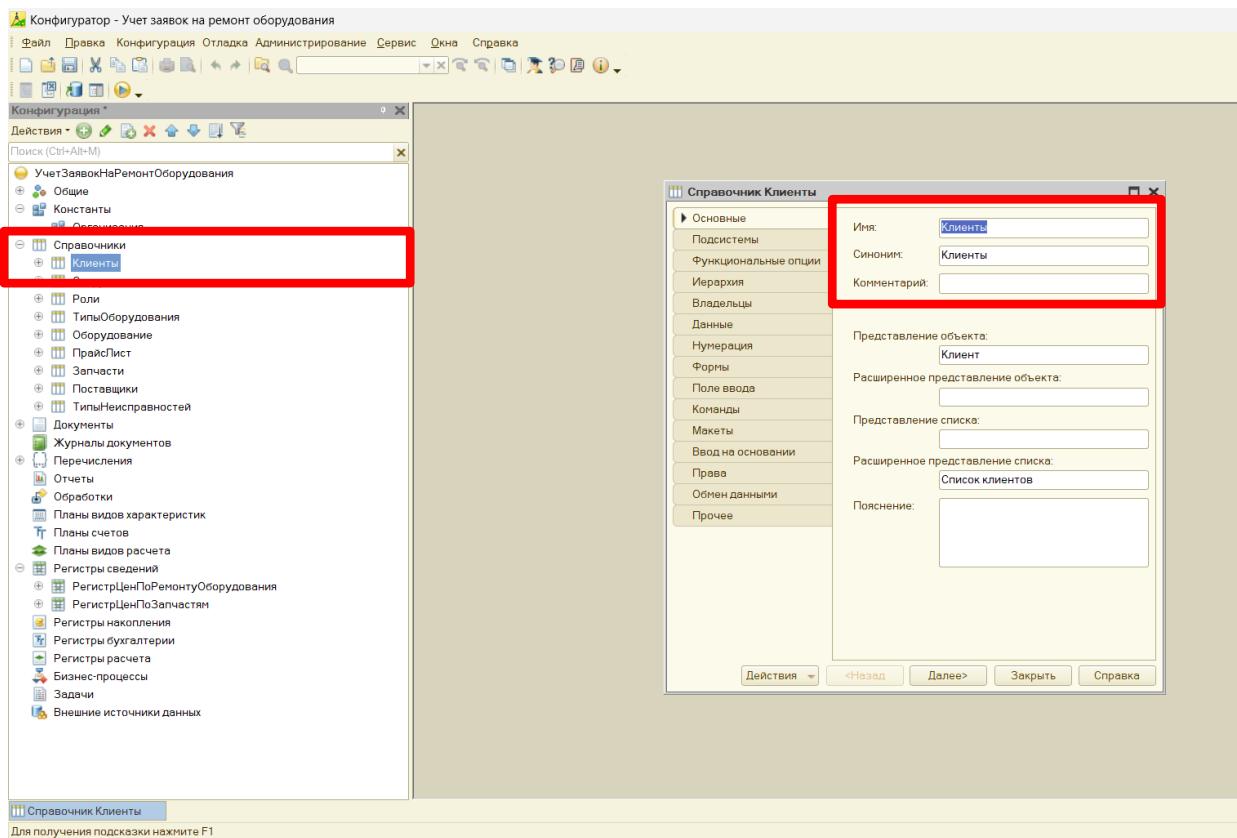
На основании имени система создаст синоним – **Роли**. Он служит для представления объекта в интерфейсе программы.



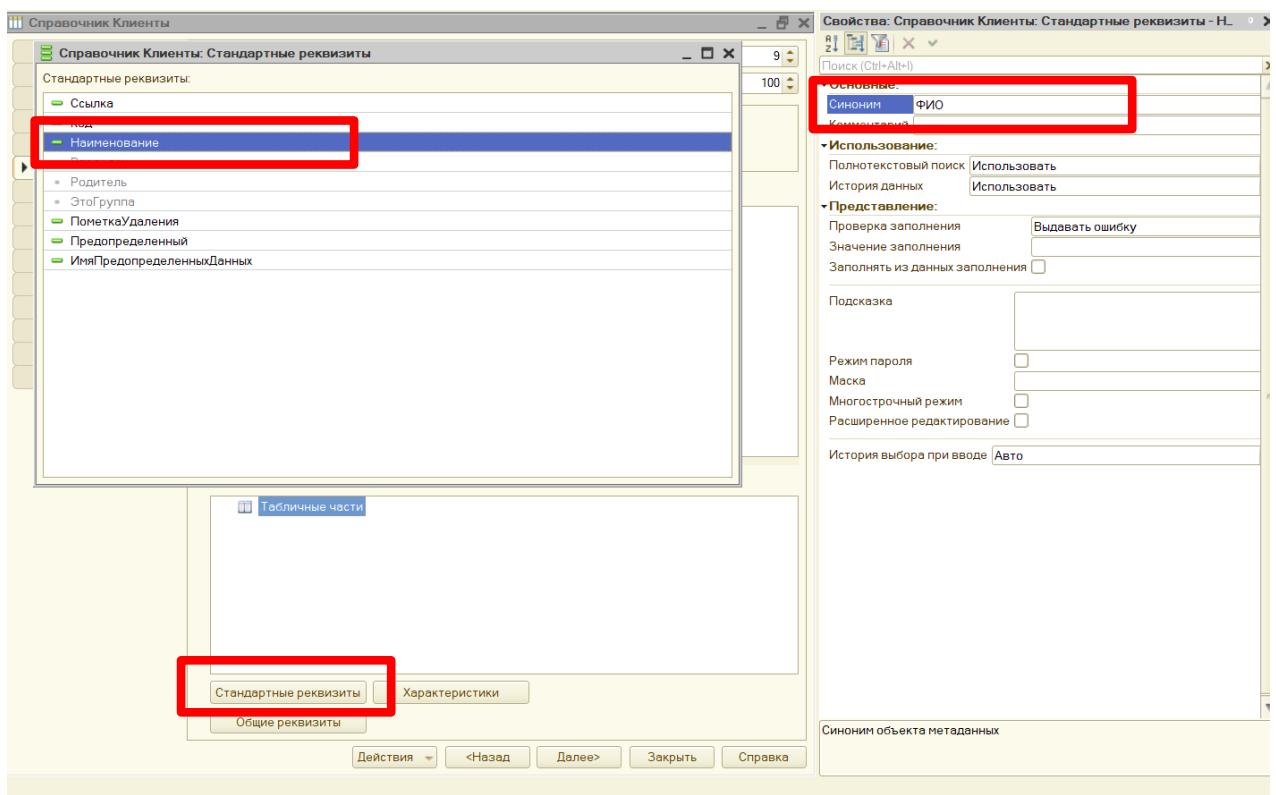
Нажмите на закладку **Данные**. Длина кода нас устраивает, а **длина наименования** оставим **25 символов**.

Запустите 1С: Предприятие в режиме пользователя внесите четыре роли: *администратор, клиент, пользователь, менеджер*.

Далее создадим объект конфигурации **Справочник – Клиенты**.



Нажмите на закладку **Данные**. Длина кода нас устраивает, а **длину наименования** установите в **100 символов**. Для наименования в **Стандартных реквизитах** мы создадим **Синоним** ФИО, в данную строку мы произведем конкатенацию реквизитов: фамилия, имя, отчество.



Добавьте в справочник следующие реквизиты:

**Фамилия** – Тип *Строка*, Длина 25;

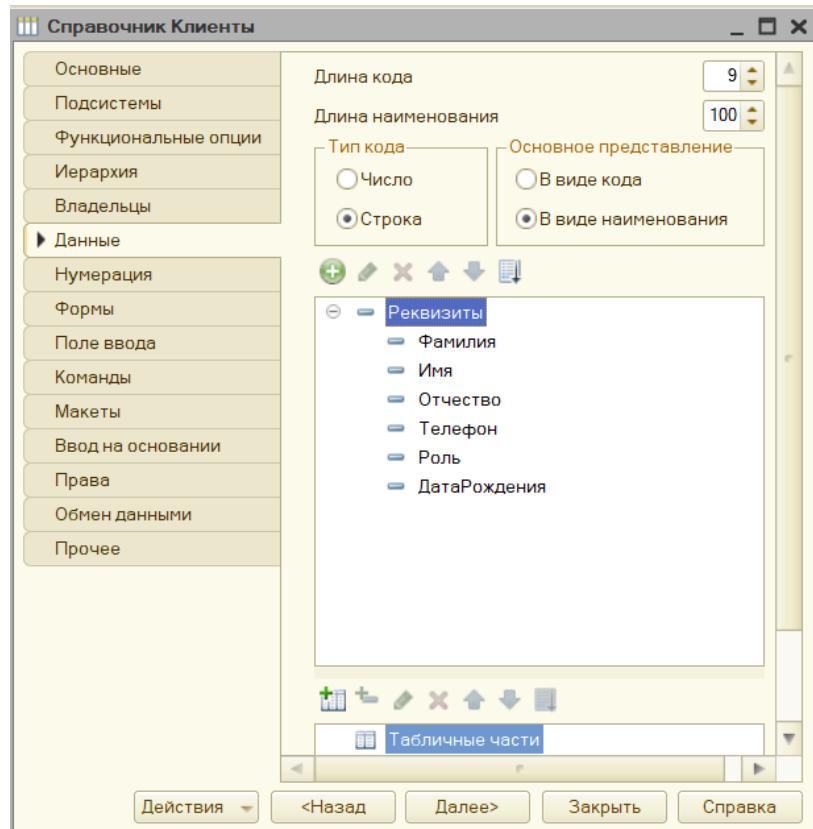
**Имя** – Тип *Строка*, Длина 25;

**Отчество** – Тип *Строка*, Длина 25;

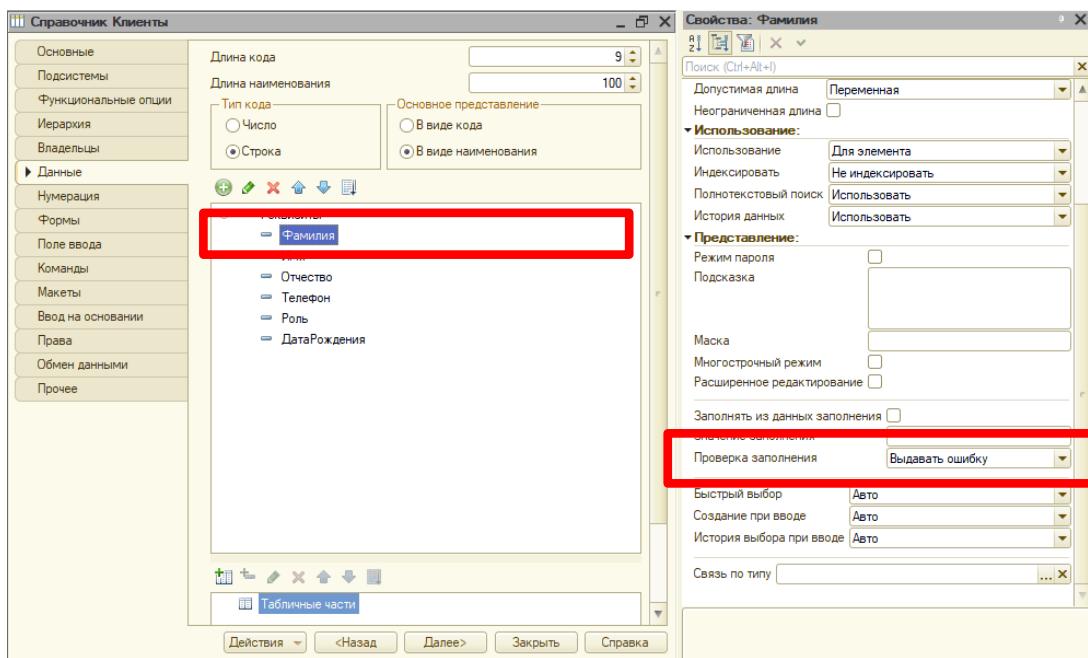
**Телефон** – Тип *Строка*, Длина 18, Маска +7 (999) 999-99-99;

**Роль** - Тип *СправочникСсылка.Роли*;

**ДатаРождения** – Тип *Дата*.



**Внимание!** При создании справочников и документов не забудьте установить обязательную проверку заполнения полей.



Согласно заданию, необходимо предусмотреть справочник ***Сотрудники***. В дереве конфигурации добавьте новый объект Справочник – ***Сотрудники***.

Нажмите на закладку **Данные**. Длина кода нас устраивает, а длину наименования установите в 100 символов. Для наименования в *Стандартных реквизитах* мы создадим *Синоним* - ***ФИО***, в данную строку мы произведем конкатенацию реквизитов: фамилия, имя, отчество.

Добавьте в справочник следующие реквизиты:

**Фамилия** – Тип *Строка*, Длина 25;

**Имя** – Тип *Строка*, Длина 25;

**Отчество** – Тип *Строка*, Длина 25;

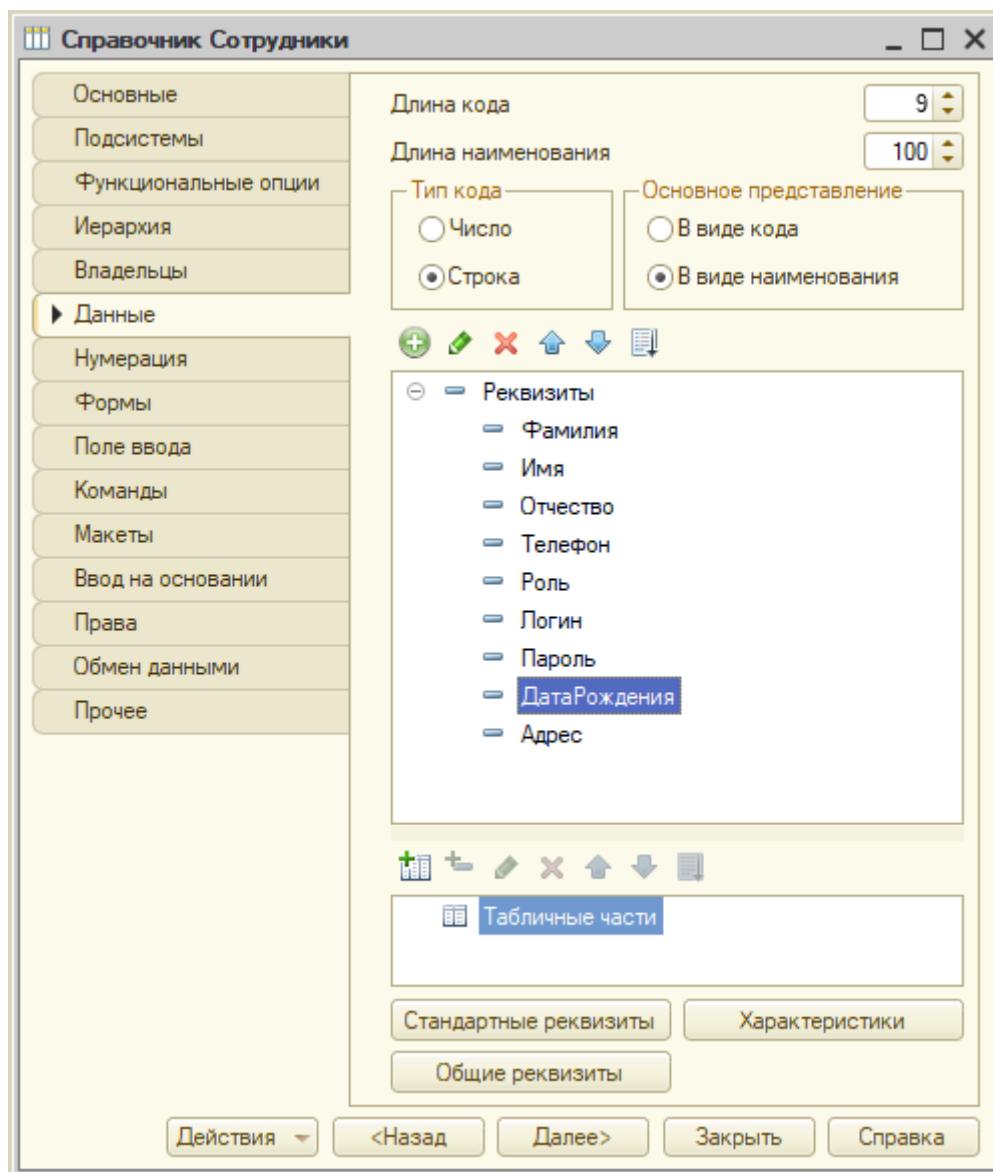
**Телефон** – Тип *Строка*, Длина 18, *Маска* +7 (999) 999-99-99;

**Роль** - Тип *СправочникСсылка.Роли*;

**Логин** – Тип *Строка* 25;

**Пароль** – Тип *Строка* 25;

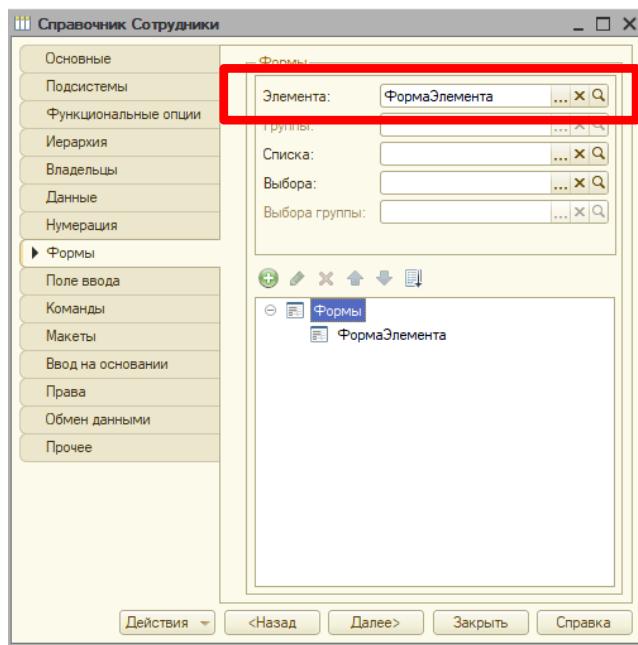
**ДатаРождения** – Тип *Дата*.



Форма сотрудника будет выглядеть следующим образом.

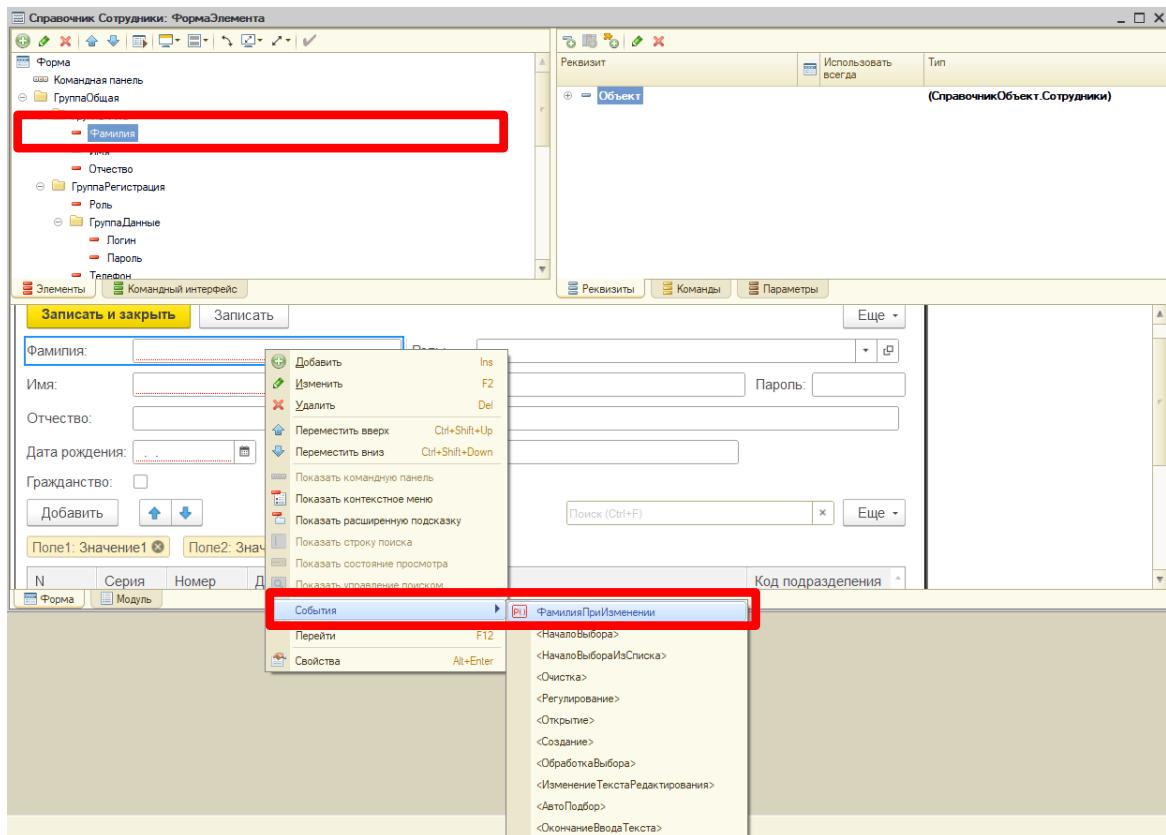
Записать и закрыть		Записать	Еще
Фамилия:	<input type="text"/>	Роль:	<input type="text"/>
Имя:	<input type="text"/>	Логин:	<input type="text"/>
Отчество:	<input type="text"/>	Пароль:	<input type="text"/>
Дата рождения:	<input type="text"/> <input type="button" value="..."/>		
Телефон:	<input type="text"/>		
Адрес:	<input type="text"/>		

В справочниках **Клиент** и **Сотрудники**, создайте форму элемента справочника, для этого на вкладке **Формы** нажмите на лупу напротив **Формы - Элемента**.



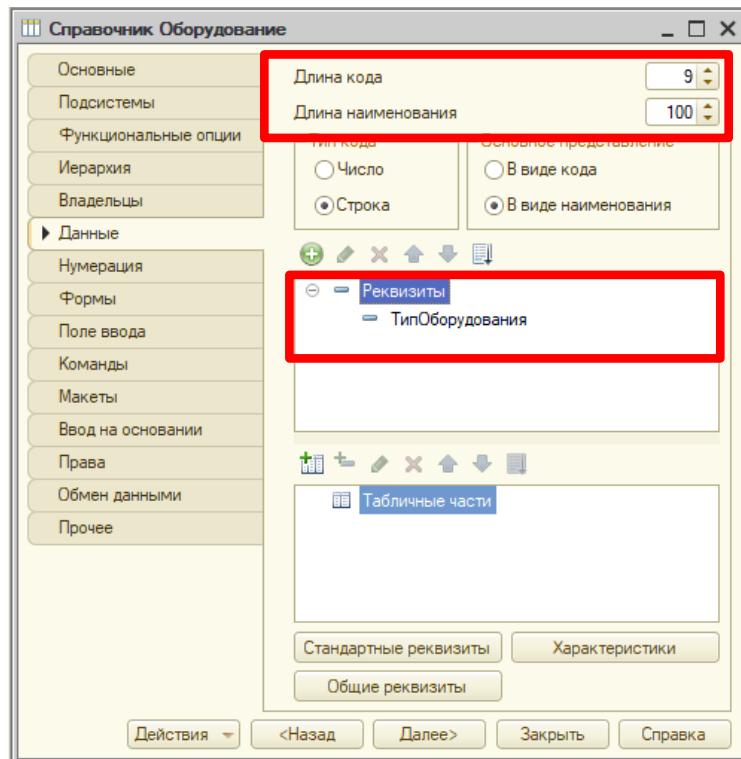
Далее выполним конкатенацию реквизитов Фамилия, Имя, Отчество. Выделите на *Форме* реквизит **Фамилия** и создайте *Действие – ПриИзменении* на *Клиенте*.

```
&НаКлиенте
Процедура ФамилияПриИзменении(Элемент)
    Объект.Наименование = Объект.Фамилия + " " + Объект.Имя + " " + Объект.Отчество;
КонецПроцедуры
```

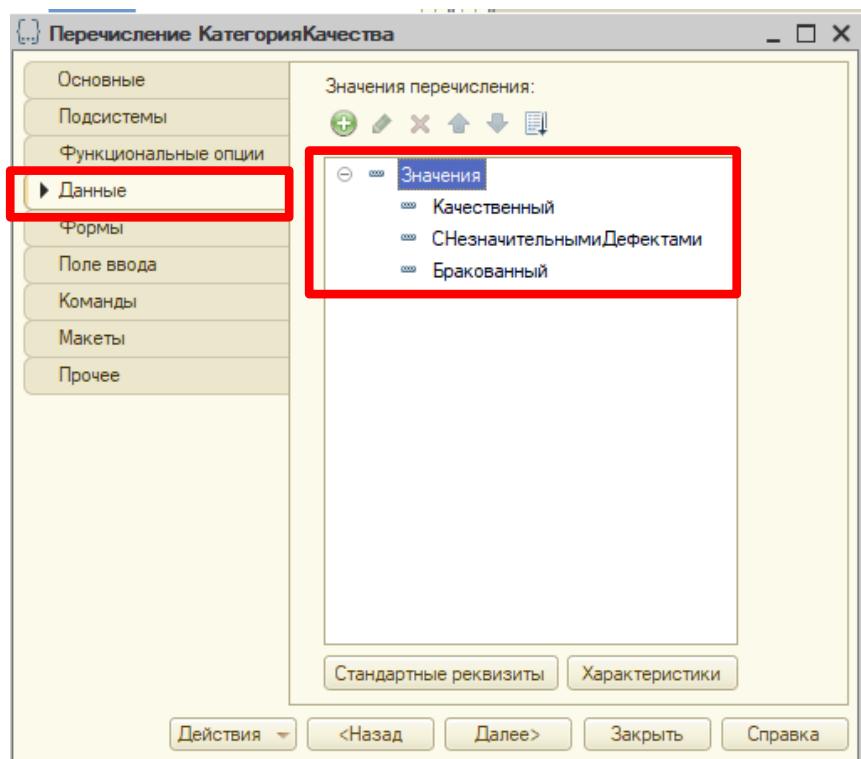
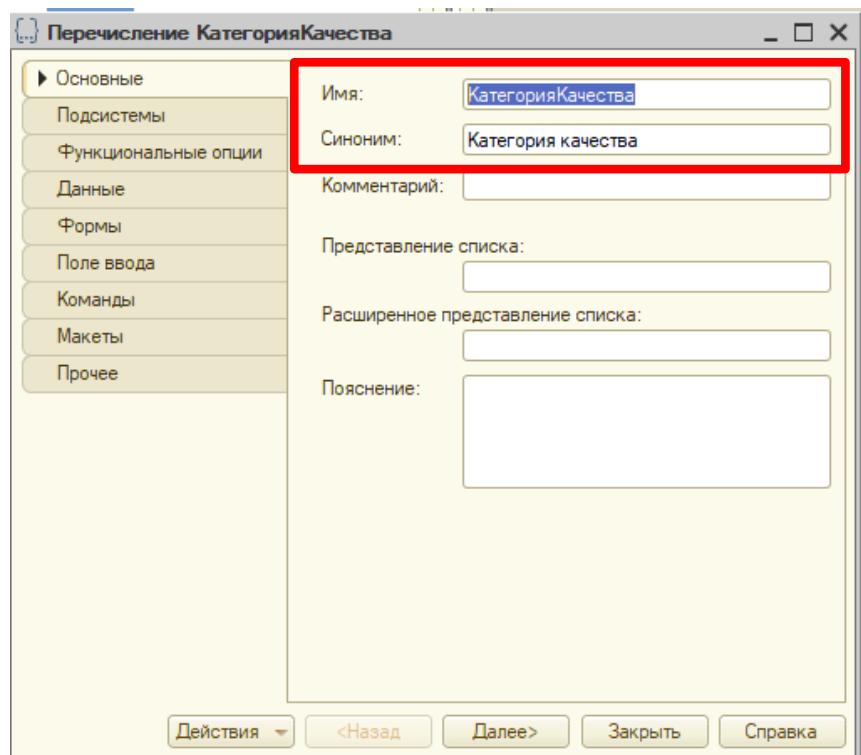


Создадим дополнительно справочники **ТипОборудования**, **ТипНеисправности**. Создаем со стандартными реквизитами **Код** и **Наименование** – Тип Стока, Длина 100.

Создадим справочник **Оборудование** со стандартными реквизитами **Код** и **Наименование** – Тип Стока, Длина 100. Добавим реквизит **ТипОборудования** – Тип СправочникСсылка.ТипыОборудования.



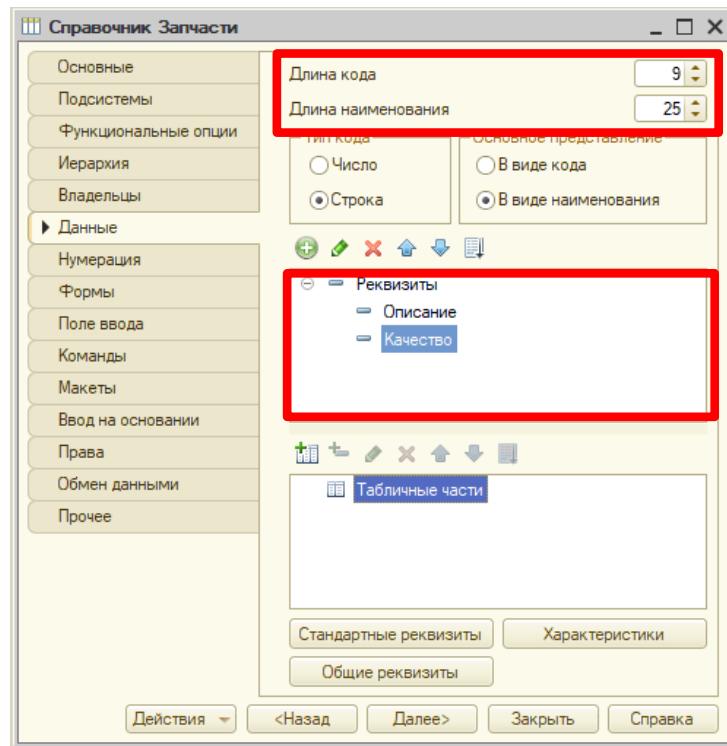
Создадим объект конфигурации **Перечисление – КатегорияКачества**. В дальнейшем мы его будем использовать в справочнике **Запчасти**. Пример представлен ниже:



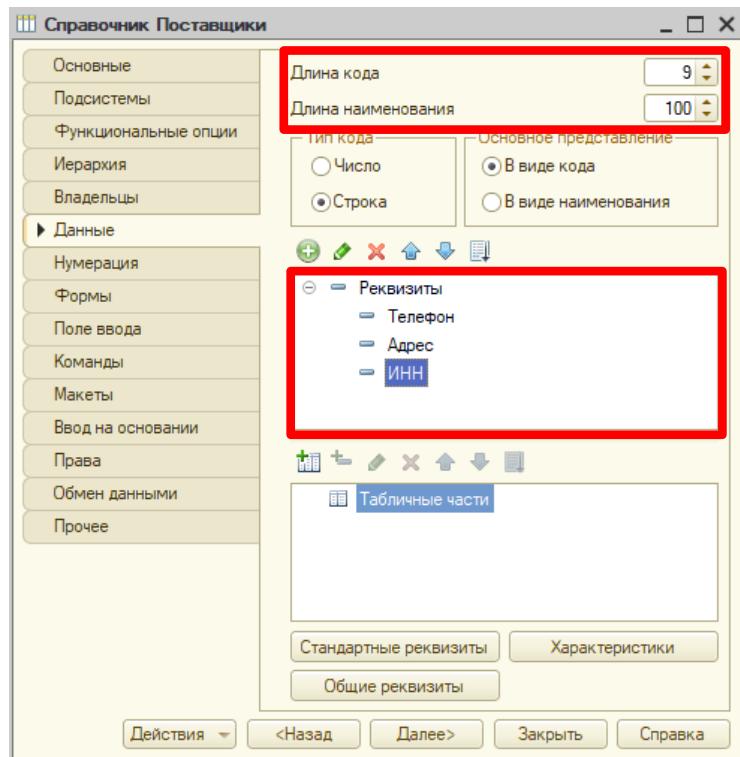
Аналогично создадим справочник **Запчасти** со стандартными реквизитами **Код** и **Наименование** – Тип *Строка*, Длина 100. Добавим дополнительных три реквизита:

<b>Описание</b>	-	Тип	<i>Строка</i>	(Неограниченная),	<b>Качество</b>	-	Тип
-----------------	---	-----	---------------	-------------------	-----------------	---	-----

Перечисление Ссылка. Категория Качества.



Аналогично создадим справочник **Поставщики** со стандартными реквизитами **Код** и **Наименование** – Тип *Строка*, Длина 100. Добавим дополнительных три реквизита: **Телефон** - Тип *Строка* - 18, **Адрес** – Тип *Строка* – 100, **ИНН** – *Строка* – 12.



Основная цель учёта заявок на ремонт оборудования - эффективное и оперативное осуществление ремонтных работ с минимизацией простоев и удовлетворением запросов

клиентов или сотрудников. Эта предметная область широко используется в различных сферах деятельности, таких как сервисные услуги, производство, информационные технологии и другие.

Предметная область учёта заявок на ремонт оборудования касается процесса подачи, обработки и учёта заявок на ремонт различного оборудования. В данной области включены следующие основные составляющие, для которых мы создадим определенные документы.

1. **Заявка на ремонт:** это информация, предоставленная клиентом или сотрудником о неисправности оборудования, которое требует ремонта. Заявка может содержать данные о типе оборудования, его серийном номере, описании проблемы и другой важной информации.

Добавьте в документ следующие реквизиты:

**Клиент** – Тип *СправочникСсылка.Клиенты*;

**Сотрудник** – Тип *СправочникСсылка.Сотрудники*;

**СтатусЗаявки** – Тип *ПеречислениеСсылка.СтатусЗаявки*;

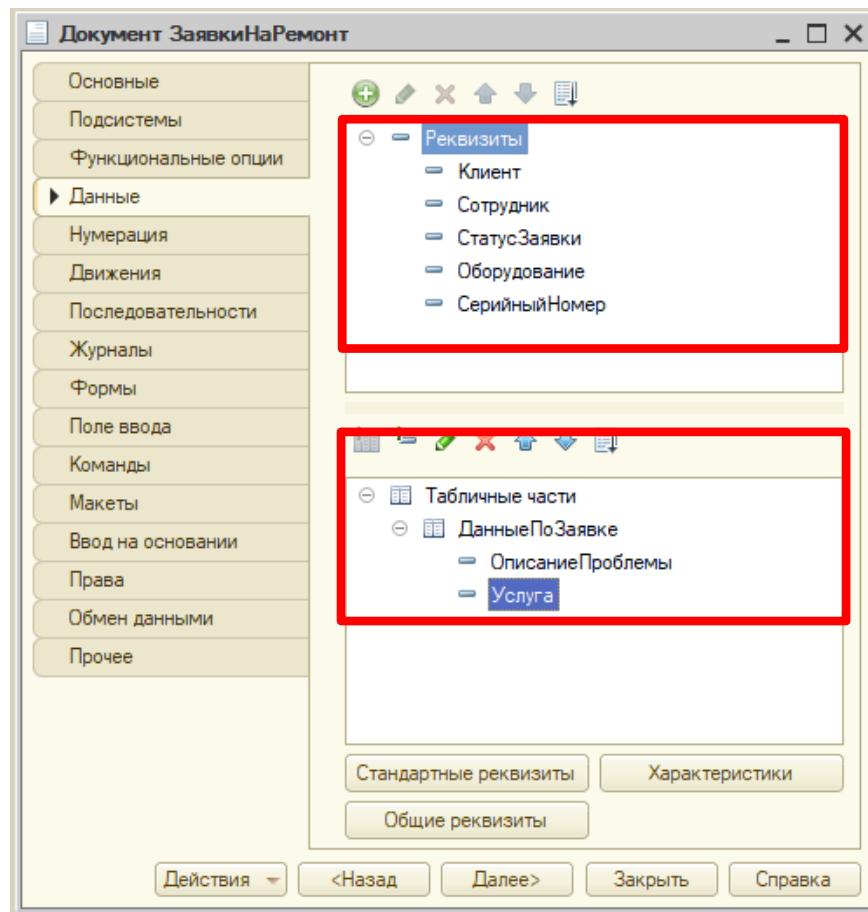
**Оборудование** – Тип *СправочникСсылка.Оборудование*;

**СерийныйНомер** – Тип *Строка*, Длина 7.

**ТабличнаяЧасть – ДанныеПоЗаявке**

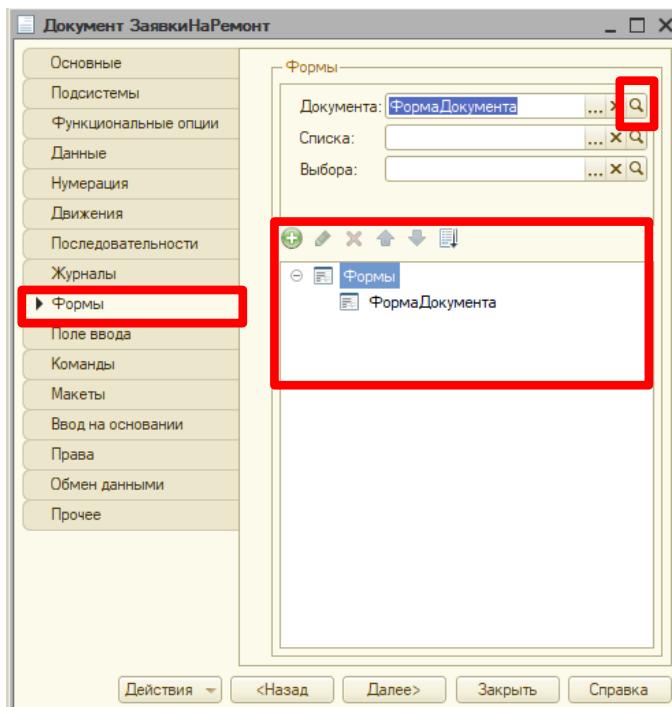
**ОписаниеПроблемы** – Тип *Строка*, Неограниченная;

**Услуга** – Тип *СправочникСсылка.Услуга*.

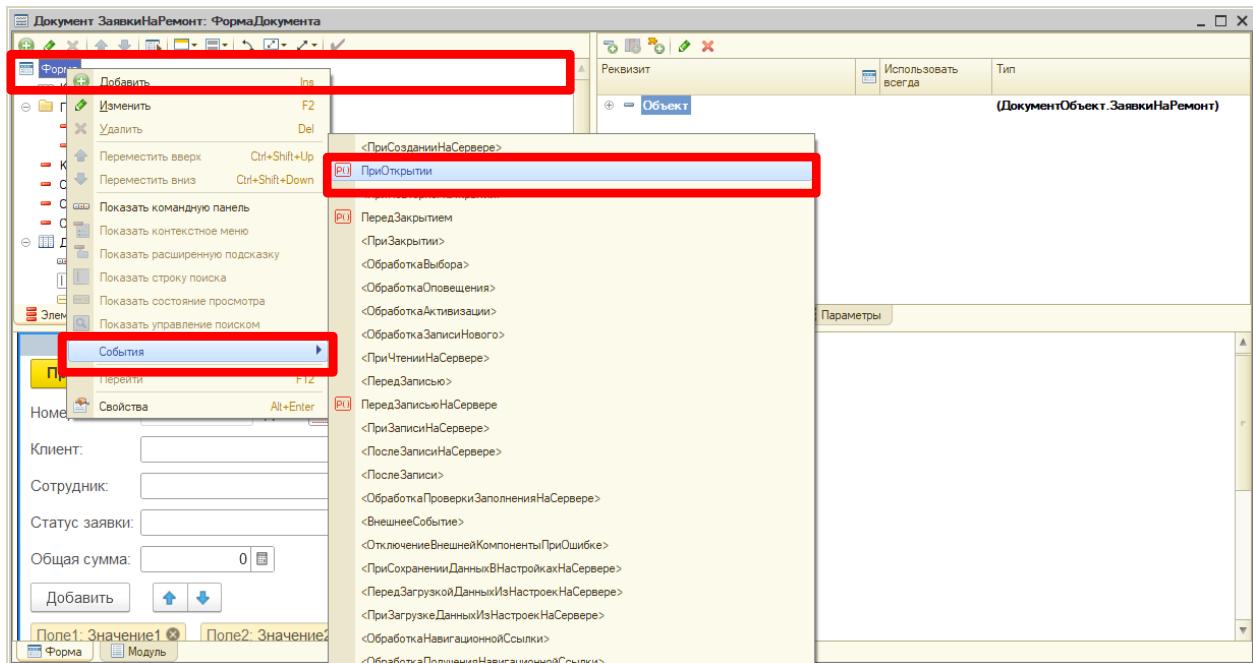


После того как Вы создали все реквизиты для документа его необходимо настроить.

Для этого создадим форму Документа, пример представлен ниже:



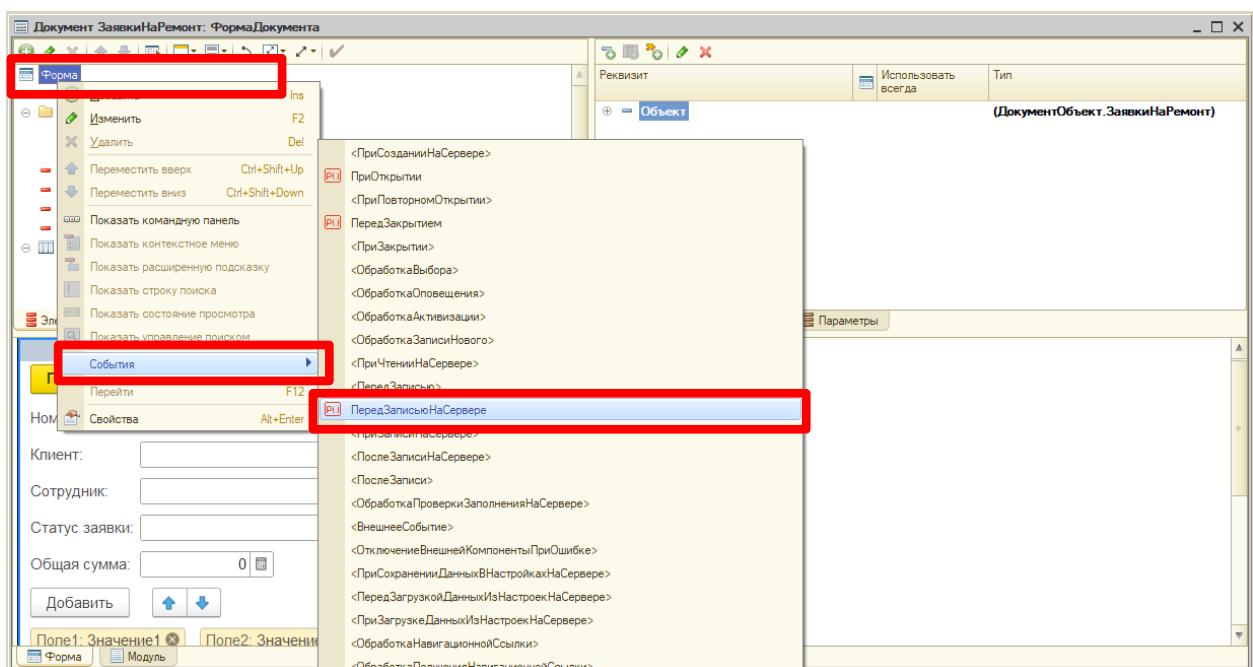
Первое событие, которое нам необходимо создать это автоматическая подстановка статуса в документе.



Для этого в форме документа *ЗаявкиНаРемонт* вызываем *Событие* при открытии и прописываем следующий код.

```
&НаКлиенте
Процедура ПриОткрытии(Отказ)
    Объект.СтатусЗаявки = ПредопределенноеЗначение ("Перечисление.СтатусЗаявки.ВОжидании")
КонецПроцедуры
```

Теперь при открытии документа у Вас будет всегда отображаться статус *В ожидании*. На следующем этапе нам необходимо создать событие на *Сервере* *ПередЗаписьюНаСервере*, которое будет выдавать сообщение пользователю о том, что нельзя изменить статус в документе *ЗаявкаНаРемонт* со статуса *ВОжидании* на другой, такая функция будет доступна только в документе *РегистрацияЗаявок*.

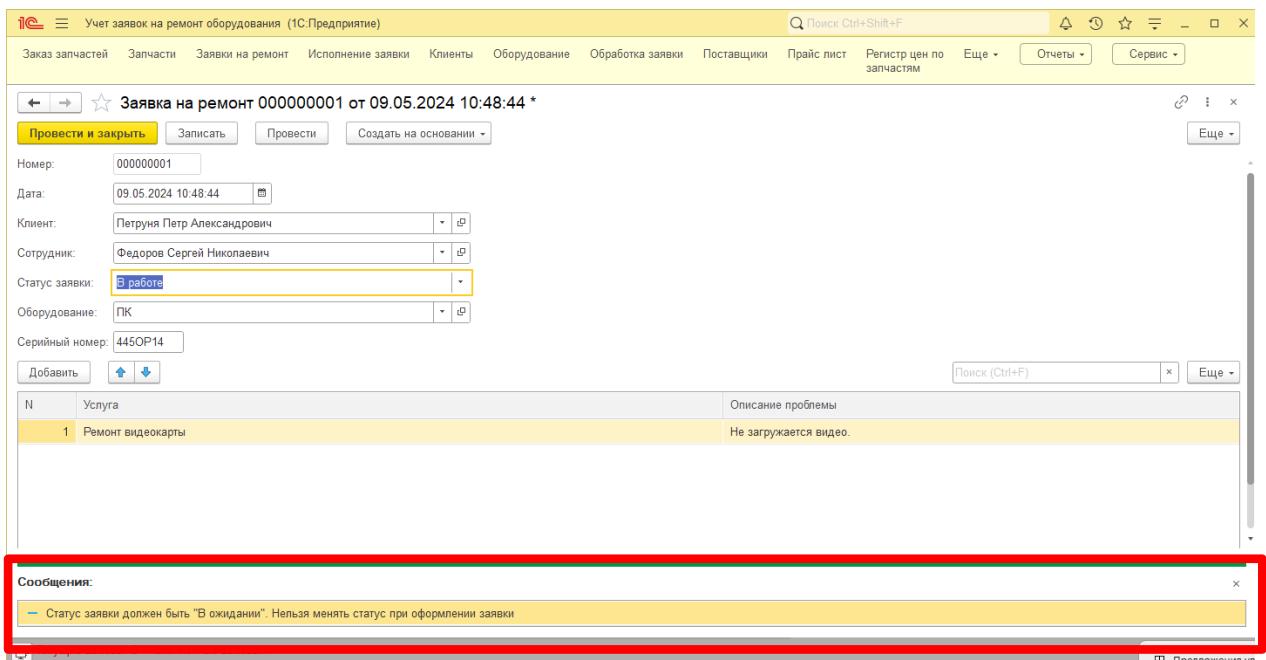


```

&НаСервере
Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)
    Если Объект.СтатусЗаявки = Перечисления.СтатусЗаявки.ВОжидании Тогда
        Объект.СтатусЗаявки = ПредопределенноеЗначение ("Перечисление.СтатусЗаявки.ВОжидании");
    Иначе
        Отказ = Истина;
        Сообщить ("Статус заявки должен быть ""В ожидании"". Нельзя менять статус при оформлении заявки");
    КонецЕсли;
КонецПроцедуры

```

Запустите платформу 1С: Предприятия, откройте документ *Заявка на ремонт* и измените статус документа и проведите документ. Если Вы все сделали правильно, то у Вас будет такое же сообщение как на примере ниже.



**2. Регистрация заявки:** этот процесс включает приём и регистрацию заявки в системе учёта. Важными аспектами регистрации являются присвоение уникального идентификатора заявке, сохранение информации о заявке и её приоритете.

Добавьте в документ следующие реквизиты:

**ДокументОснования** – Тип *ДокументСсылка.ЗаявкаНаРемонт*;

**Клиент** – Тип *СправочникСсылка.Клиенты*;

**Сотрудник** – Тип *СправочникСсылка.Сотрудники*;

**СтатусЗаявки** – Тип *ПеречислениеСсылка.СтатусЗаявки*;

**Оборудование** – Тип *СправочникСсылка.Оборудование*;

**СерийныйНомер** – Тип *Строка*, Длина 7;

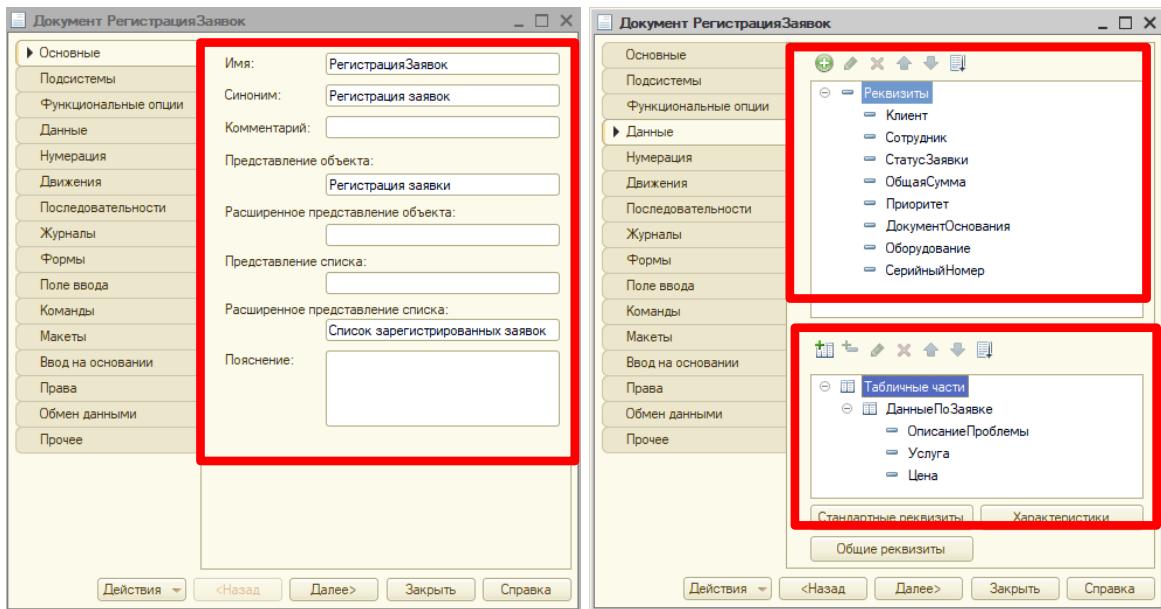
**ОбщаяСумма** – Тип Число, Длина 10, Точность 2, Неотрицательное;

**ТабличнаяЧасть – ДанныеПоЗаявке**

**ОписаниеПроблемы** – Тип *Строка*, Длина Неограниченная;

**Услуга** – Тип *СправочникСсылка.Услуга*;

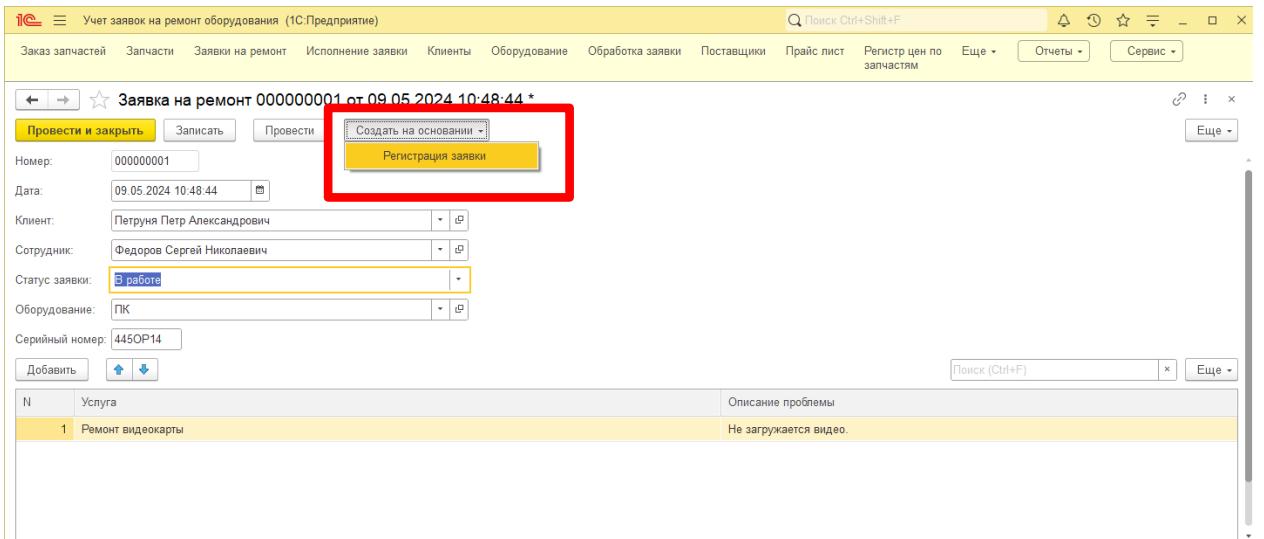
**Цена – Тип Число, Длина 10, Точность 2, Неотрицательное;**



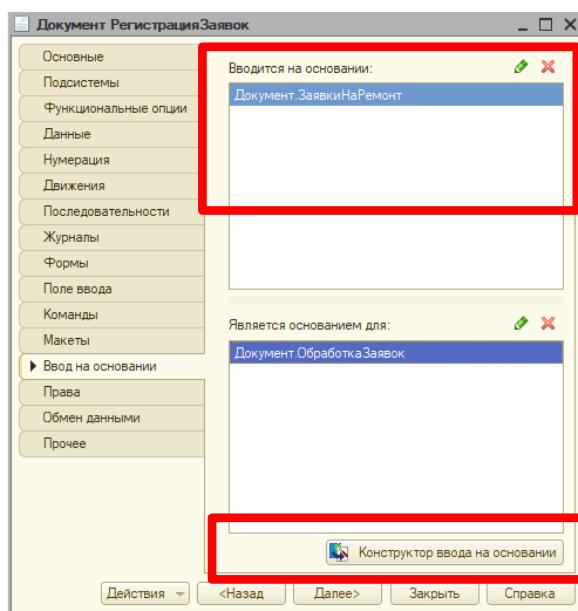
После того как Вы создали все реквизиты для документа его необходимо настроить. Для этого создадим форму *Документа*.

После того, как клиент оформит **Заявку на ремонт оборудования**, сотруднику следует оформить документ **Регистрация заявки**, в котором указать оборудование и примерную сумму, которую ему необходимо клиенту необходимо будет оплатить за ремонт оборудования. Поэтому документ должен быть создан на основании **Заявки на ремонт**.

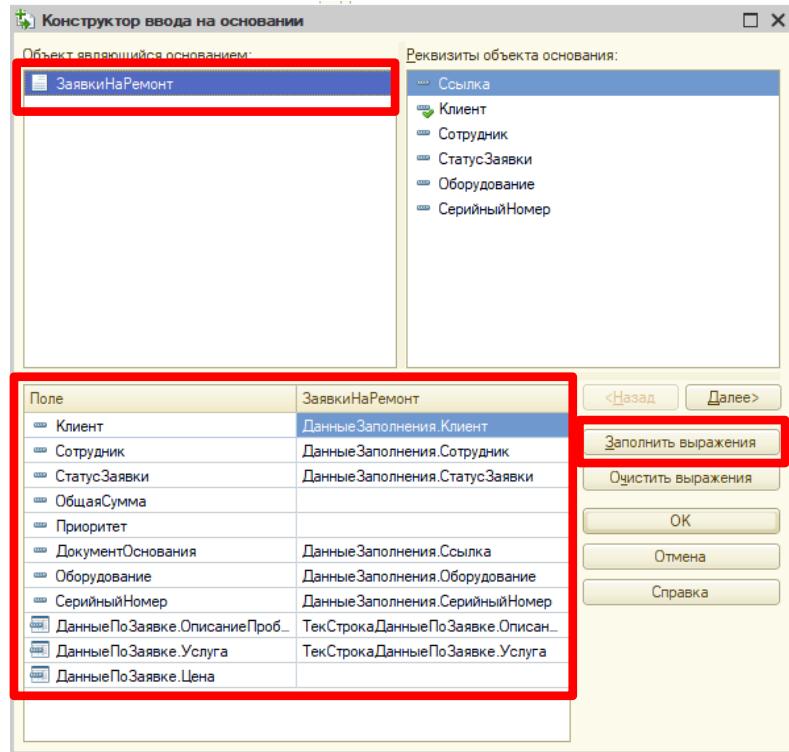
В такой ситуации механизм ввода на основании позволяет сформировать документ **Регистрация заявки** автоматически, используя информацию, которая хранится в документе **Заявку на ремонт**.



Открываем документ **Регистрация заявки**, вкладка *Ввод на основании* далее выбираем документ **ЗаявкаНаРемонт**, далее необходимо сформировать **Конструктор ввода на основании** помогает создать процедуру на встроенном языке.



Конструктор позволяет вручную или автоматически заполнить выражения, которые будут записаны в поля результирующего объекта прикладного решения:



Результатом работы конструктора является готовая процедура на встроенным языке с именем *ОбработкаЗаполнения()*. Эта процедура располагается в модуле результирующего объекта конфигурации и будет вызвана системой при вводе этого объекта на основании другого объекта.

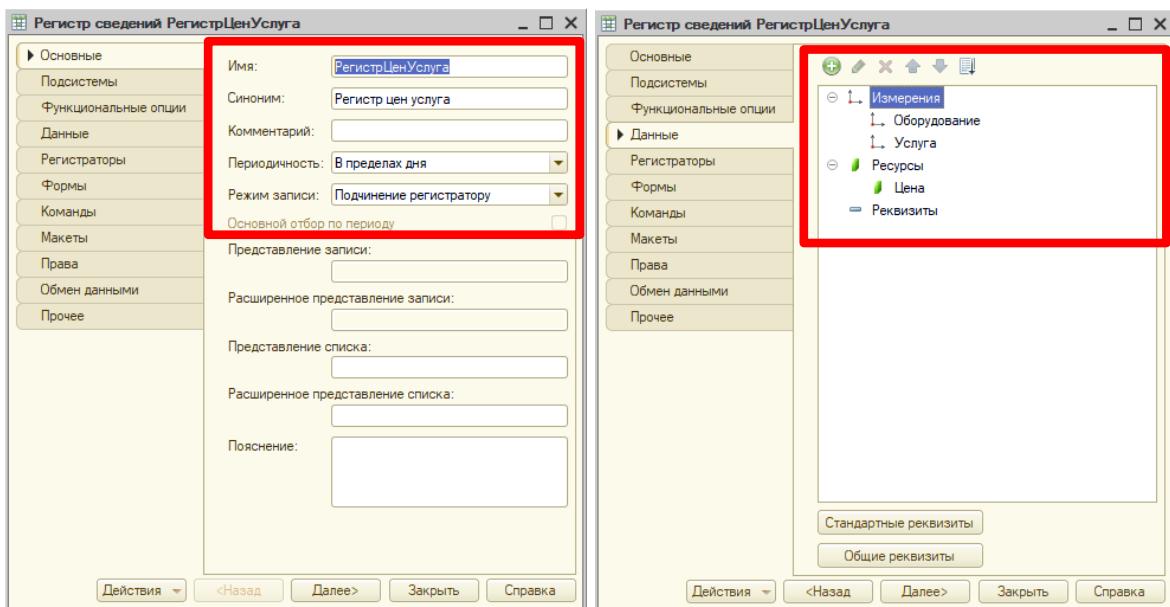
```

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
//{{ __КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
Если ТипВнч(ДанныеЗаполнения) = Тип("ДокументСсылка.ЗаявкиНаРемонт") Тогда
    // Заполнение шапки
    Клиент = ДанныеЗаполнения.Клиент;
    Оборудование = ДанныеЗаполнения.Оборудование;
    СерийныйНомер = ДанныеЗаполнения.СерийныйНомер;
    Сотрудник = ДанныеЗаполнения.Сотрудник;
    ДокументОснования = ДанныеЗаполнения.Ссылка;
    СтатусЗаявки = ДанныеЗаполнения.СтатусЗаявки;
    Для Каждого ТекСтроДанныеПоЗаявке Из ДанныеЗаполнения.ДанныеПоЗаявке Цикл
        НоваяСтрока = ДанныеПоЗаявке.Добавить();
        НоваяСтрока.ОписаниеПроблемы = ТекСтроДанныеПоЗаявке.ОписаниеПроблемы;
        НоваяСтрока.Услуга = ТекСтроДанныеПоЗаявке.Услуга;
    КонецЦикла;
    КонецЕсли;
//}}__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры

```

Запустите платформу 1С: Предприятие и посмотрите, как заполниться документ *Регистрация заявки*, на основании документа *Заявка на ремонт*.

Теперь наша задача запрограммировать форму таким образом, чтобы при выборе оборудования у нас в зависимости от типа неисправности производилась автоматическая цена и подсчитывалась общая сумма. Для этого создадим *РегистрСведений – РегистрЦенУслуга*. Настройте его согласно примеру, ниже:



На вкладке *Данные* создадим реквизиты, по которым мы будем отслеживать изменение цен.

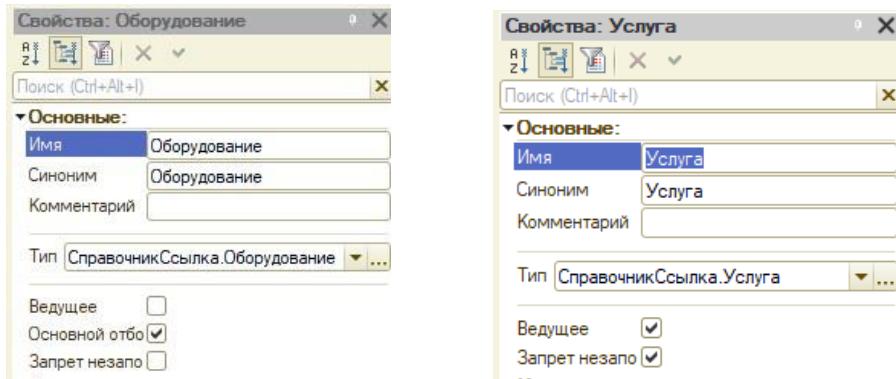
Добавьте в документ следующие реквизиты:

**Оборудование** – Тип *СправочникСсылка.Оборудование*;

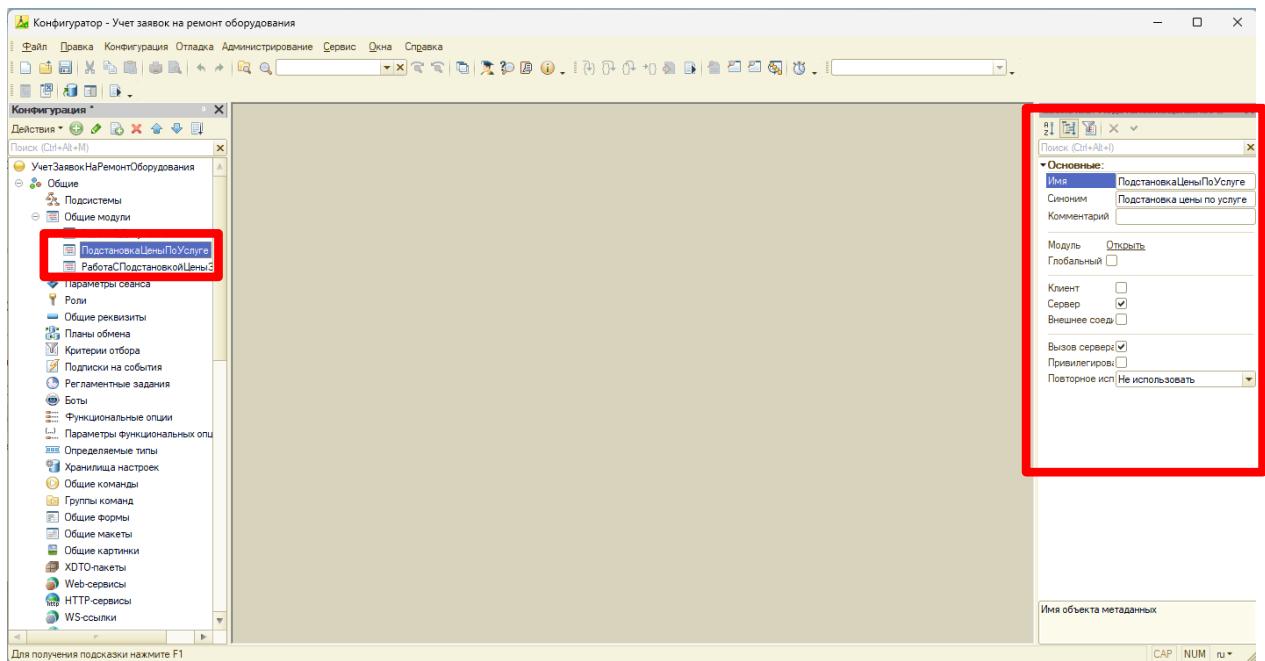
**Услуга** – Тип *СправочникСсылка.Услуга*;

**Цена** – Тип *Число*, Длина 10, Точность 2, Неотрицательное.

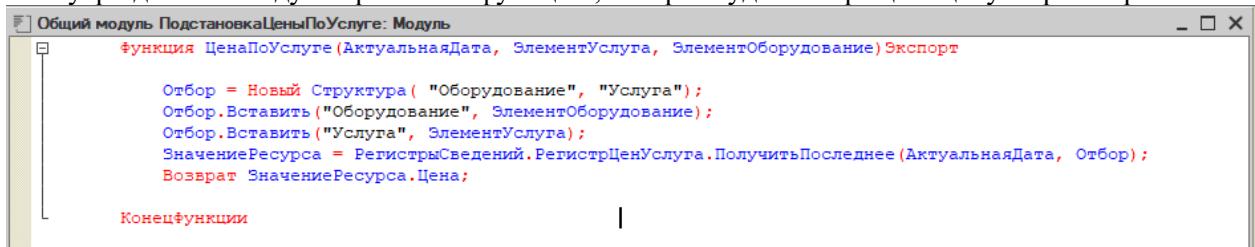
В свойствах реквизитов проверьте наличие следующих галочек.



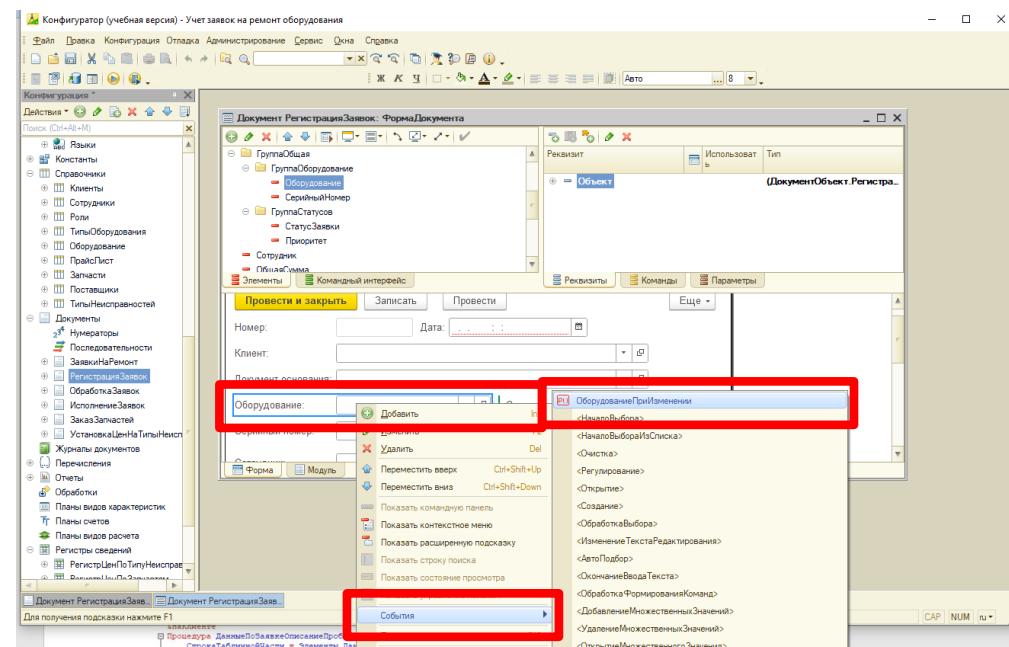
На следующем шаге Вам необходимо чтобы в табличной части документа при выборе *Описаний проблемы* у нас автоматически подставлялась цена из регистра *РегистрЦенУслуга* табличную часть. Для этого мы создадим с Вами общий модуль с названием *ПодстановкаЦеныПоУслуге*, пример представлен ниже:



Внутри данного модуля пропишем функцию, которая будет возвращать цену из регистра.



Теперь можно перейти к автоматизации ТЧ документа. В форме документа выделяем реквизит Оборудование и создаем событие **ПриИзменении**.



Также создадим событие **ПриИзменении** на реквизит табличной части Услуга. Наша задача заключается в том, чтобы при выборе **Оборудования** цена по **Услуге** изменялась в соответствии с регистром. Поэтому мы создадим вручную процедуру **ПересчетСтроки** и ее запишем в процедуры,

которые мы создали.

```
«НаКлиенте
└ Процедура СтатусЗаявкиПриИзменении(Элемент)
    └ СтатусЗаявкиПриИзмененииНаСервере();
└ КонецПроцедуры

«НаКлиенте
└ Процедура ОборудованиеПриИзменении(Элемент)
    └ ПересчетСтроки();
└ КонецПроцедуры
«НаКлиенте
└ Процедура ПересчетСтроки()

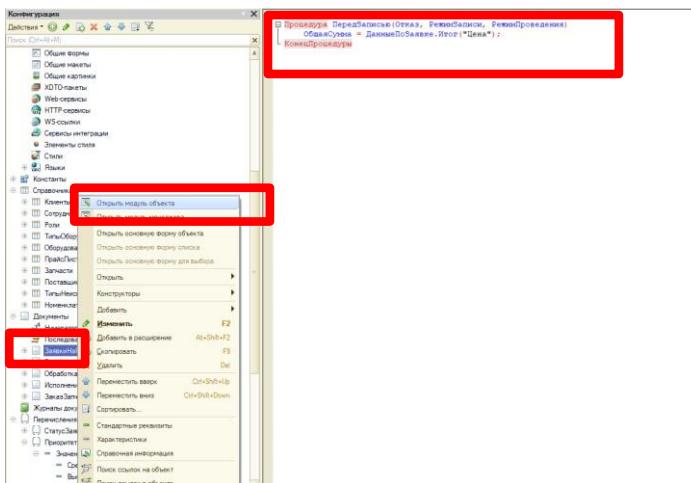
    СтрокаТабличнойЧасти = Элементы.ДанныеПоЗаявке.ТекущиеДанные;
    СтрокаТабличнойЧасти.Цена = ПодстановкаЦеныПоУслуге(Объект.Дата, СтрокаТабличнойЧасти.Услуга, Объект.Оборудование);

└ КонецПроцедуры

«НаКлиенте
└ Процедура ДанныеПоЗаявкеУслугаПриИзменении(Элемент)
    └ ПересчетСтроки(); КонецПроцедуры
```

Обратите внимание, что в табличной части необходимо реализовать подвал, для этого воспользуйтесь инструкцией с сайта <https://dzen.ru/a/YJout0-t43iL6csT>

Чтобы на форме документа после заполнения табличной части в реквизит **ОбщаяСумма** подала *Общая цена* из подвала вашего документа, необходимо открыть Модуль объекта и в нем прописать *Процедуру подстановки*.



**3. Обработка заявки:** процесс, включающий анализ заявки, определение её приоритетности и назначение исполнителя (ремонтного специалиста) для задачи. В процессе обработки может потребоваться дополнительная информация или уточнение деталей проблемы у клиента или сотрудника.

### **ВНИМАНИЕ!!!**

*Данный документ создаётся по аналогии предыдущего примера*, вплоть до автоматизации. Вы можете предыдущий документ скопировать переименовать, и тогда вся автоматизация и элементы программирования будут в нем присутствовать, останется только добавить в него недостающие реквизиты.

Добавьте в документ следующие реквизиты:

**ДокументОснования** - Тип *ДокументСсылка.РегистрацияЗаявок*

**Клиент** – Тип *СправочникСсылка.Клиенты*;

**Сотрудник** – Тип *СправочникСсылка.Сотрудники*;

**СтатусЗаявки** – Тип *ПеречислениеСсылка.СтатусЗаявки*;

**ОбщаяСумма** – Тип *Число*, Длина 10, Точность 2, Неотрицательное;

**Оборудование** – Тип *СправочникСсылка.Оборудование*;

**СерийныйНомер** – Тип *Строка*, Длина 7;

**Приоритет** – Тип *ПеречислениеСсылка.Приоритет*; Для данного реквизита необходимо создать объект перечисление **Приоритет** со значениями – средний, высокий, низкий.

**ДополнительнаяИнформация** – Тип *Строка*, Длина 255;

**ДатаНачалаРемонта** – Тип *Дата*;

**ДатаОкончанияРемонта** – Тип *Дата*;

**ТабличнаяЧасть – ДанныеПоЗаявке**

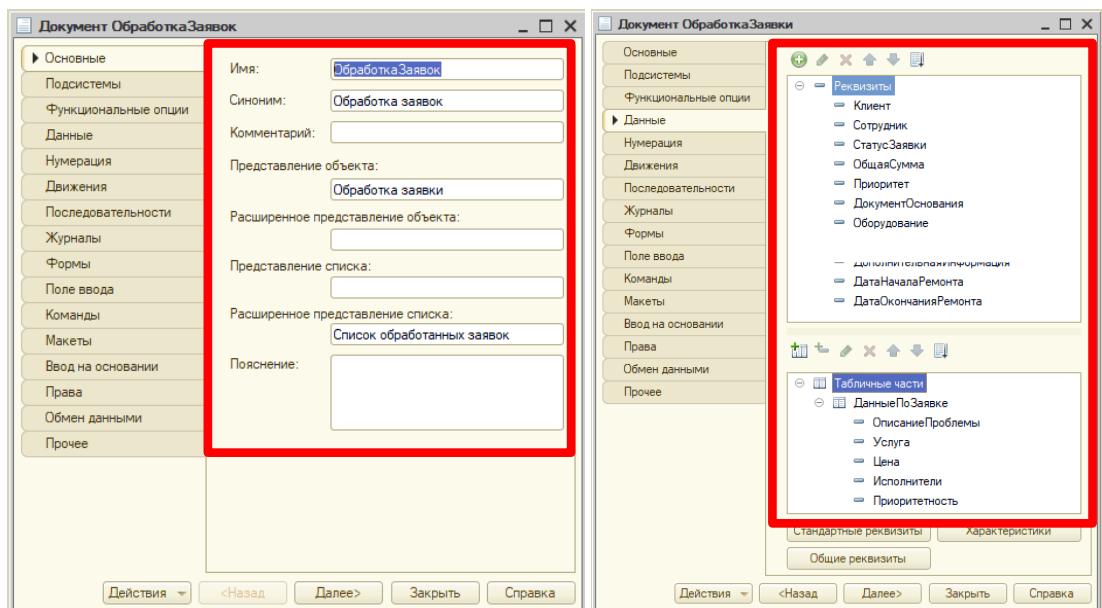
**ОписаниеПроблемы** – Тип *Строка*, Неограниченная;

**Услуга** – Тип *СправочникСсылка.Услуга*;

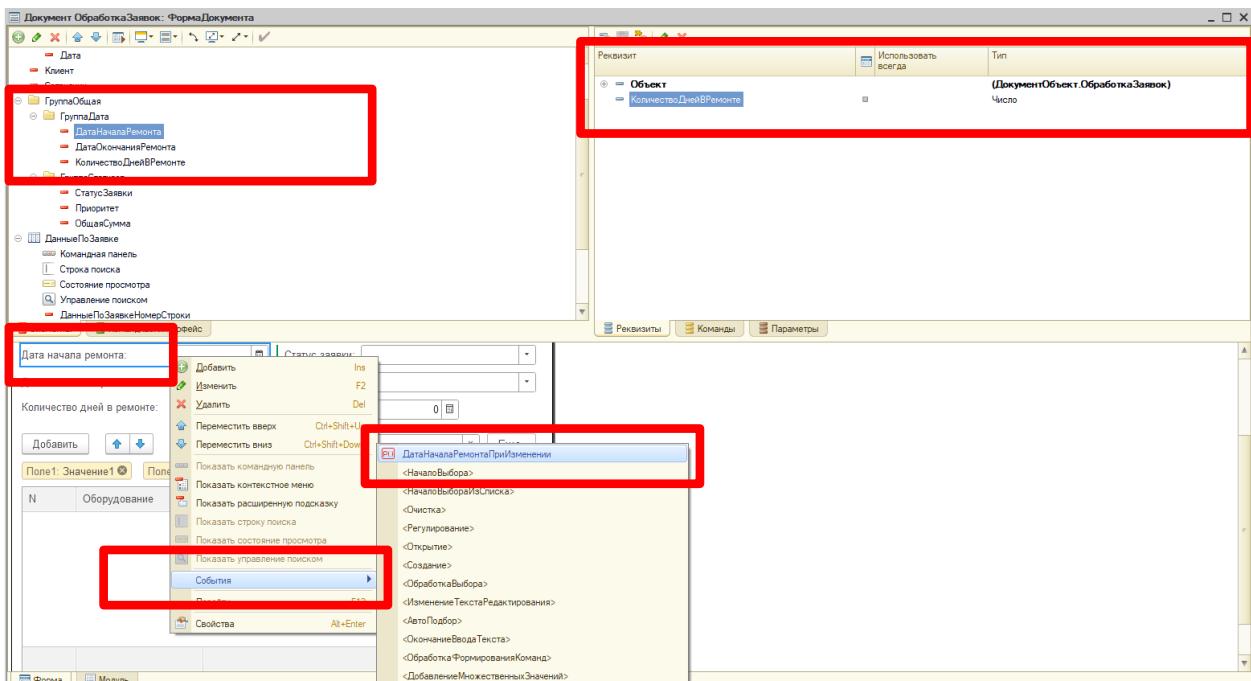
**Цена** – Тип *Число*, Длина 10, Точность 2, Неотрицательное;

**Исполнители** – Тип *СправочникСсылка.Сотрудники*;

**Приоритет** – Тип *ПеречислениеСсылка.Приоритет*.



Ввод на основании создается по аналогии предыдущим документом, только в данном случае Вы его уже создаете на документ **РегистрацияЗаявок**. В данном документе необходимо рассчитать сколько дней оборудование будет находиться в ремонте, для этого создадим событие **ПриИзменении** на Дату начала и Окончания ремонта, пример представлен ниже:



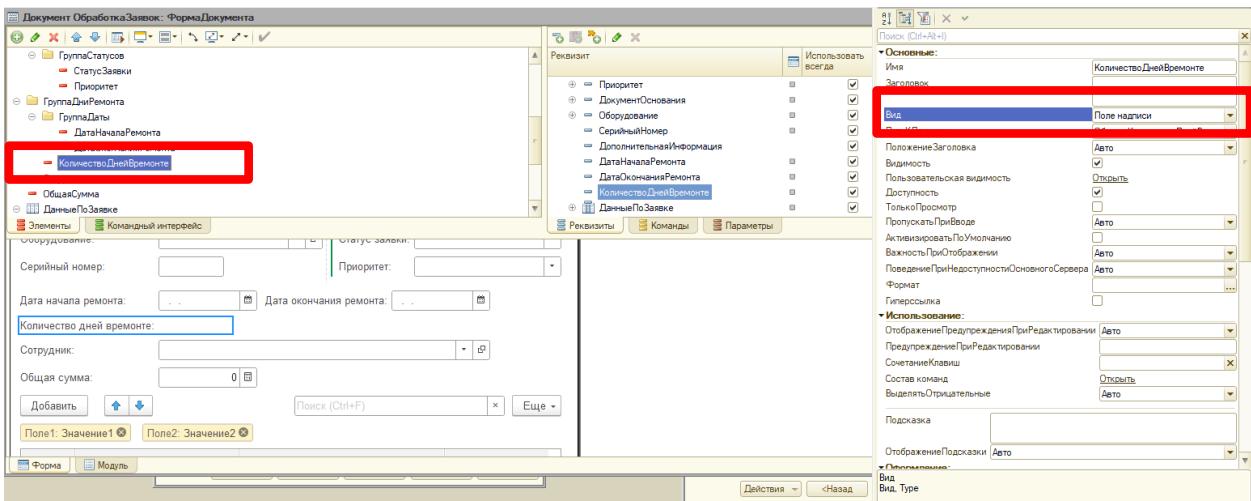
```

&НаКлиенте
Процедура ДатаНачалаРемонтаПриИзменении(Элемент)
    ПосчитатьДниРемонта();
КонецПроцедуры

&НаКлиенте
Процедура ДатаОкончанияРемонтаПриИзменении(Элемент)
    ПосчитатьДниРемонта();
КонецПроцедуры

&НаКлиенте
Процедура ПосчитатьДниРемонта()
    Если ЗначениеЗаполнено(Объект.ДатаНачалаРемонта) И ЗначениеЗаполнено(Объект.ДатаОкончанияРемонта) Тогда
        НачалоДня1 = НачалоДня(Объект.ДатаНачалаРемонта);
        НачалоДня2 = КонецДня(Объект.ДатаОкончанияРемонта) + 1;
        РазницаСекунд = НачалоДня2 - НачалоДня1;
        КоличествоДнейВРемонте = РазницаСекунд / (24 * 3600);
    КонецЕсли;
КонецПроцедуры
  
```

Чтобы у Вас результат записался в реквизит Количество дней в ремонте, Вам необходимо на форме Вид изменить на поле надписи.



Запустите платформу 1С: Предприятие и проверьте все ли данные подставляются и заполняются.

**4. Исполнение заявки:** фактическое выполнение ремонта оборудования. В этом этапе назначенный исполнитель ремонтирует оборудование, вносит необходимые изменения или заменяет неисправные компоненты. Важно отметить, что на этом этапе могут возникать необходимость заказа запчастей или координации работ с другими специалистами.

**ВНИМАНИЕ!!!**

*Данный документ создаётся по аналогии предыдущего примера, вплоть до автоматизации. Вы можете предыдущий документ скопировать переименовать, и тогда вся автоматизация и элементы программирования будут в нем присутствовать, останется только добавить в него недостающие реквизиты. Только в данном документе будет уже две табличные части.*

Добавьте в документ следующие реквизиты:

**ДокументОснования** – Тип *ДокументСсылка.РегистрацияЗаявок*

**Клиент** – Тип *СправочникСсылка.Клиенты*;

**Сотрудник** – Тип *СправочникСсылка.Сотрудники*;

**СтатусЗаявки** – Тип *ПеречислениеСсылка.СтатусЗаявки*;

**ОбщаяСумма** – Тип Число, Длина 10, Точность 2, Неотрицательное;

**Оборудование** – Тип *СправочникСсылка.Оборудование*;

**СерийныйНомер** – Тип Стока, Длина 7;

**Приоритет** – Тип *ПеречислениеСсылка.Приоритет*;

**ДополнительнаяИнформация** – Тип Стока, Длина 255;

**ДатаНачалаРемонта** – Тип Дата;

**ДатаОкончанияРемонта** – Тип Дата;

**ТабличнаяЧасть – ДанныеПоЗаявке**

**ОписаниеПроблемы** – Тип Стока, Неограниченная;

**Услуга** – Тип *СправочникСсылка.Услуга*;

**Цена** – Тип Число, Длина 10, Точность 2, Неотрицательное;

**Исполнители** – Тип *СправочникСсылка.Сотрудники*;

**Приоритет** – Тип *ПеречислениеСсылка.Приоритет*.

**ТабличнаяЧасть – НеисправныеКомпоненты**

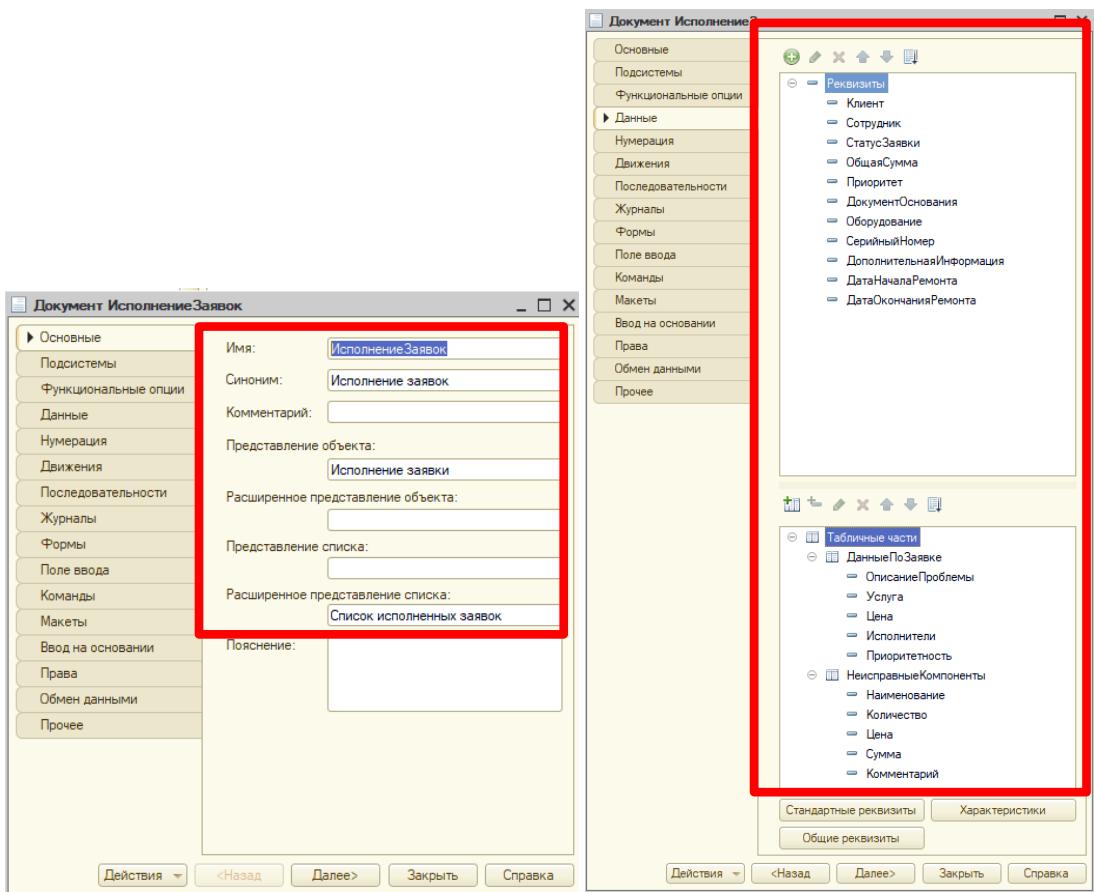
**Наименование** – Тип *СправочникСсылка.Запчасти*;

**Количество** – Тип Число, Длина 10, Точность 2, Неотрицательное;

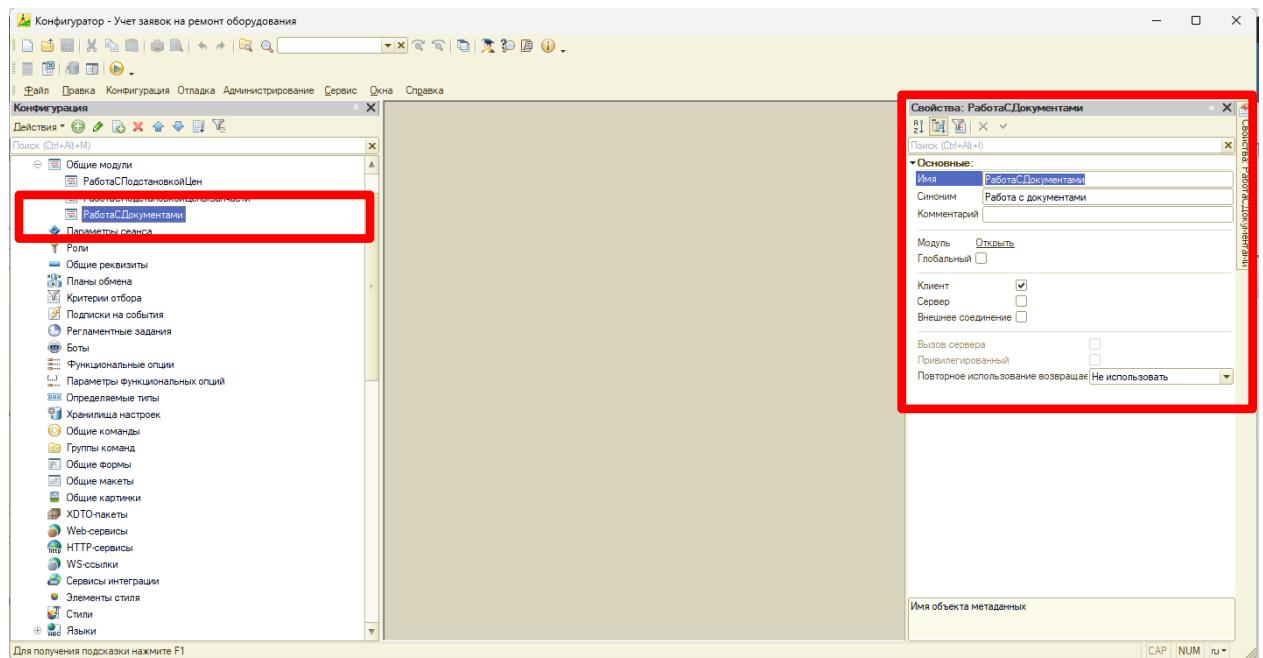
**Цена** – Тип Число, Длина 10, Точность 2, Неотрицательное;

**Сумма** – Тип Число, Длина 10, Точность 2, Неотрицательное;

**Комментарий** – Тип Стока, Длина 255.



Создадим регистр, в котором будет производиться операции расчета ТЧ нашего документа, назовем его *Работа С Документами*.



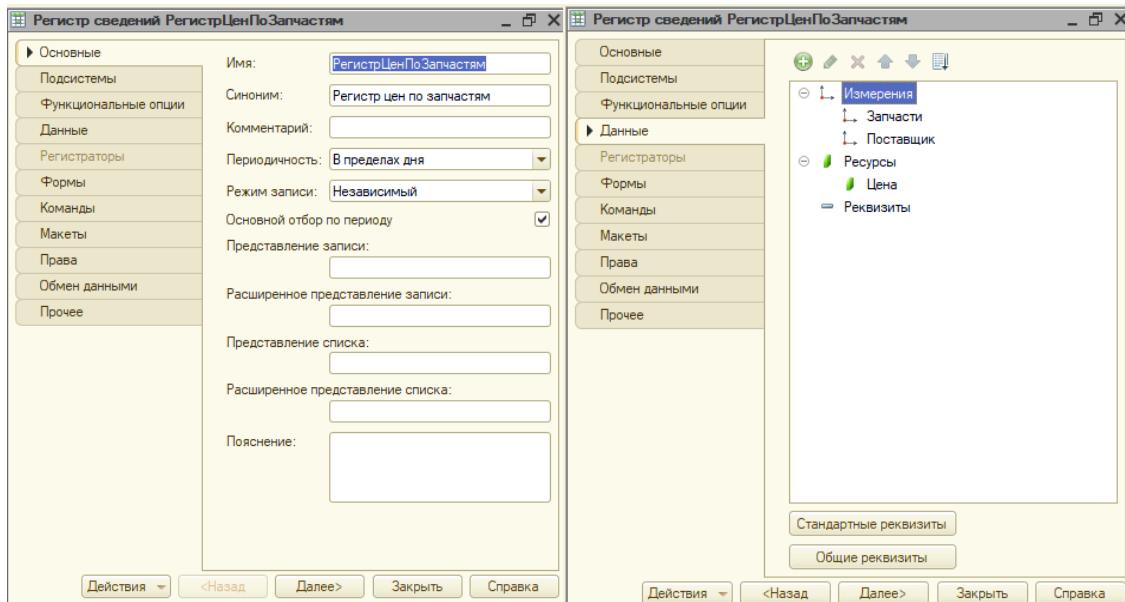
Внутри данного модуля пропишем процедуру, которая будет цену умножать на количество и возвращать итоговую стоимость.

```

Общий модуль РаботаСДокументами: Модуль
Процедура РассчитатьСумму(СтроКаТабличнойЧасти) Экспорт
СтроКаТабличнойЧасти.Сумма = СтроКаТабличнойЧасти.Цена * СтроКаТабличнойЧасти.Количество;
КонецПроцедуры

```

В данном документе в Табличной части при выборе *Запчасти* необходимо предусмотреть автоматическую подстановку цены из регистра сведений. Для этого создадим *РегистрСведений – РегистрЦенПоЗапчастям*. Настройте его согласно примеру, ниже:



На вкладке *Данные* создадим реквизиты, по которым мы будем отслеживать изменение цен.

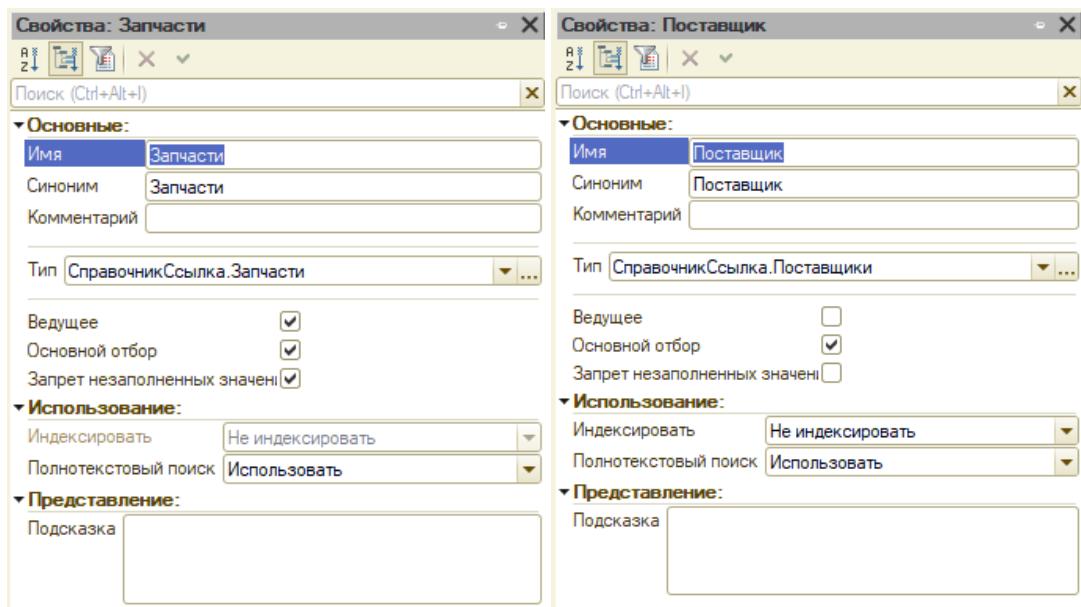
Добавьте в документ следующие реквизиты:

**Запчасти** – Тип *СправочникСсылка.Запчасти*;

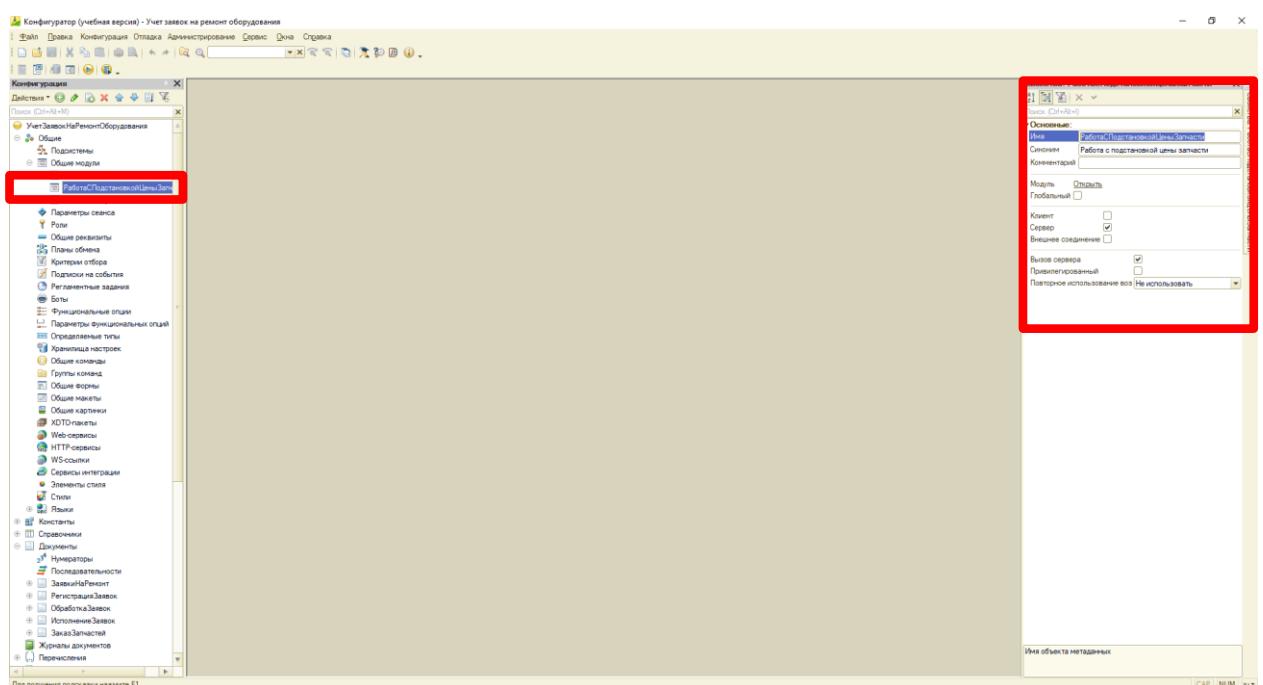
**Поставщик** – Тип *СправочникСсылка.Поставщики*;

**Цена** – Тип *Число, Длина 10, Точность 2, Неотрицательное*;

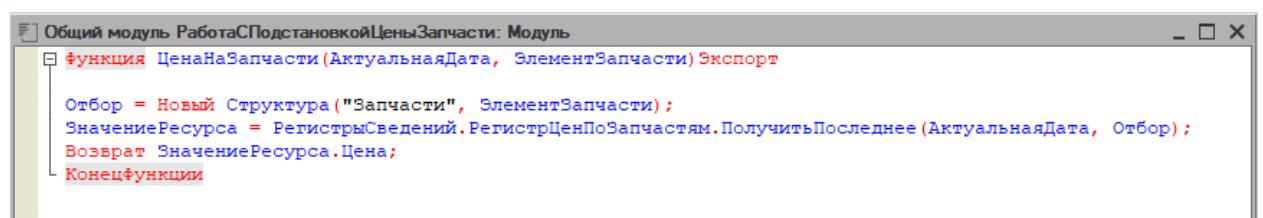
В свойствах реквизитов проверьте наличие следующих галочек.



На следующем шаге Вам необходимо чтобы в табличной части документа при выборе *Запчасти* у нас автоматически подставлялась цена из регистра *РегистрЦенПоЗапчастям* в табличную часть. Для этого мы создадим с Вами общий модуль с названием *РаботаСПодстановкойЦеныЗапчасти*, пример представлен ниже:



Внутри данного модуля пропишем функцию, которая будет возвращать цену из регистра

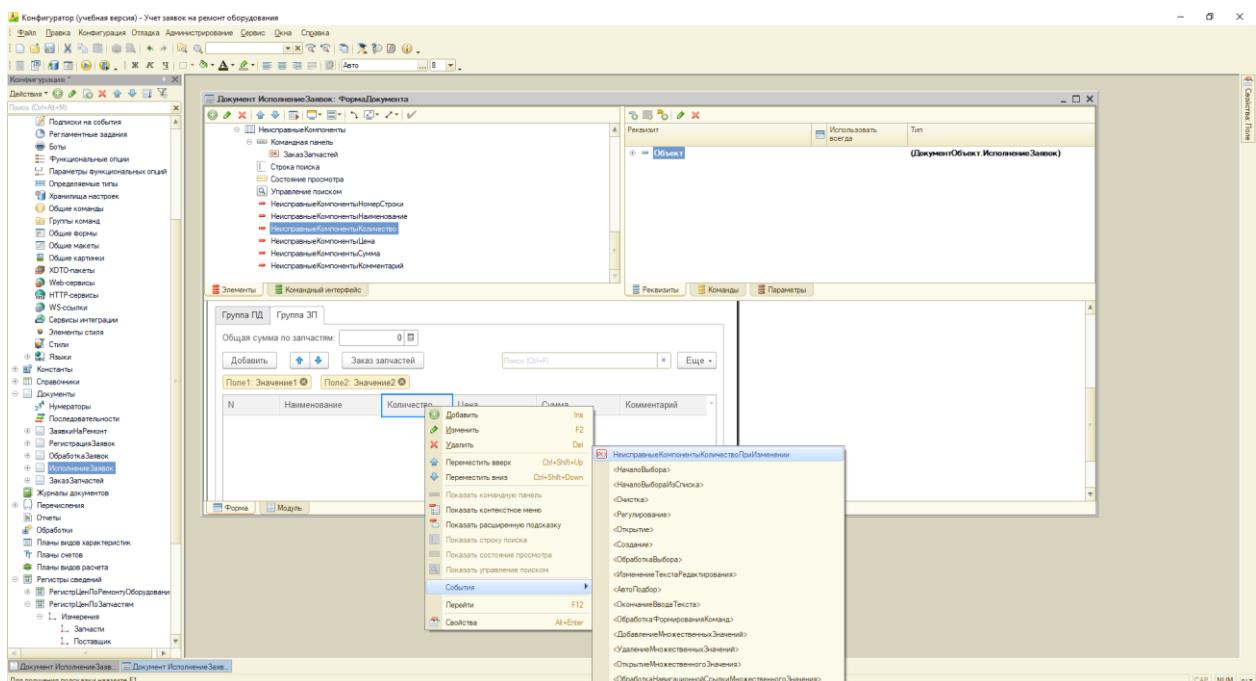


Теперь переходим к настройке ТЧ *Неисправные Компаненты* нашего документа. Как

и в предыдущих примерах нам необходимо при выборе запчасти подставить цену из регистра для этого создадим событие ПриИзменении как делали раньше. Пример кода ниже:

```
&НаКлиенте
Процедура НеисправныеКомпонентыНайменованиеПриИзменении(Элемент)
    СтрокаТабличнойЧасти = Элементы.НеисправныеКомпоненты.ТекущиеДанные;
    СтрокаТабличнойЧасти.Цена = РаботаСПодстановкойЦеныЗапчасти.ЦенаНазапчасти(Объект.Дата, СтрокаТабличнойЧасти.Найменование);
    СтрокаТабличнойЧасти.Количество = 1;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры
```

Теперь создадим событие ПриИзменении на столбцы количество и цена, внутренне содержание у процедуры будет одинаковое.



### Процедуры изменения:

```
&НаКлиенте
Процедура НеисправныеКомпонентыКоличествоПриИзменении(Элемент)
    СтрокаТабличнойЧасти = Элементы.НеисправныеКомпоненты.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры

&НаКлиенте
Процедура НеисправныеКомпонентыНайменованиеПриИзменении(Элемент)
    СтрокаТабличнойЧасти = Элементы.НеисправныеКомпоненты.ТекущиеДанные;
    СтрокаТабличнойЧасти.Цена = РаботаСПодстановкойЦеныЗапчасти.ЦенаНазапчасти(Объект.Дата, СтрокаТабличнойЧасти.Найменование);
    СтрокаТабличнойЧасти.Количество = 1;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры

&НаКлиенте
Процедура НеисправныеКомпонентыЦенаПриИзменении(Элемент)
    СтрокаТабличнойЧасти = Элементы.НеисправныеКомпоненты.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры
```

Проверяем все ли в приложении в ТЧ поставляется и рассчитывается.

Теперь нам необходимо обе ТЧ сложить и вывести сумму в реквизит Итоговая стоимость. Для этого в Дереве объектов метаданных выделяем документ **ИсполнениеЗаявки** правой кнопкой мыши выбираем **Открыть модуль объекта** и

прописываем следующую процедуру:

```
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
    ОбщаяСумма = ДанныеПоЗаявке.Итог("Цена") + НеисправныеКомпоненты.Итог("Цена");
КонецПроцедуры
```

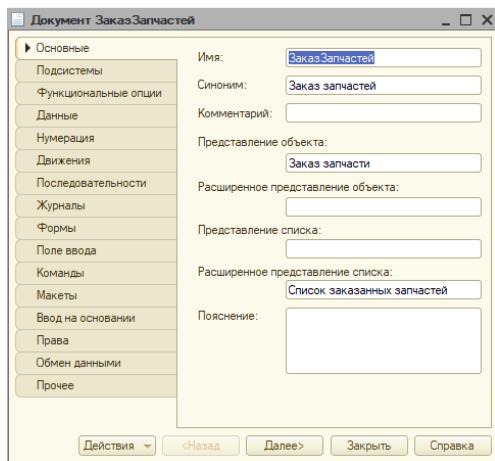
В данном документе необходимо учитывать, что, если у нас нет в наличие определенных типов запчастей мы их можем заказать у поставщика. Поэтому в документе создается дополнительная ТЧ, в которой будет фиксироваться факт использования дополнительных запчастей, заказанных для клиента.

The screenshot shows the 'Execution of application' window for application number 000000003 on 09.05.2024 at 22:45:34. The 'General information' tab is selected. The 'Total amount' field contains '4 500,00'. A red box highlights this field. Below it, the 'Information about parts' tab is open, showing a single part entry: '1 Термостат TF308' with a quantity of 1 and a unit price of 3 000,00.

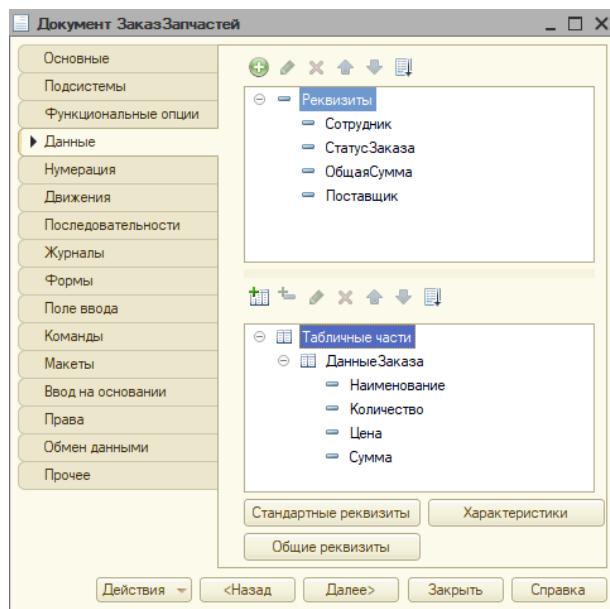
The screenshot shows the 'Execution of application' window for application number 000000003 on 09.05.2024 at 22:45:34. The 'General information' tab is selected. The 'Total amount' field contains '4 500,00'. A red box highlights this field. Below it, the 'Information about parts' tab is open, showing a single part entry: '1 Не работает' (Not working) with a service item 'Замена лампочки' (Replace bulb) and a responsible employee 'Стреколовский Сергей Анатольевич'.

Как уже говорилось ранее у нас могут отсутствовать запчасти на ремонт оборудования поэтому необходимо предусмотреть документ **ЗаказЗапчастей** и у поставщика. Автоматизацию данного примера выполнить самостоятельно по примеру

документа *ИсполнениеЗаказа*.



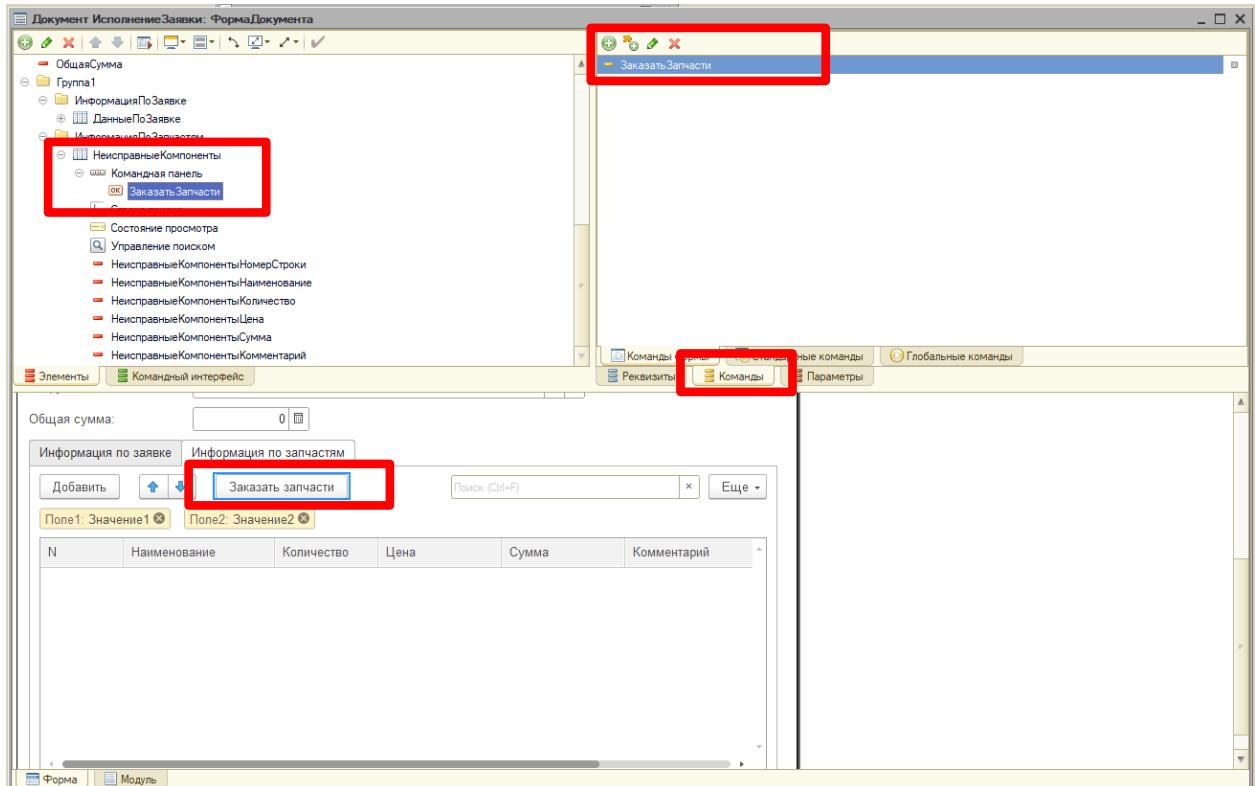
Типы для реквизитов определите самостоятельно



Настройте форму документа и запрограммируйте соответствующие элементы

На форму документа *ИсполнениеЗаявки* добавьте кнопку *Заказать запчасти у поставщика*,

пример ниже:

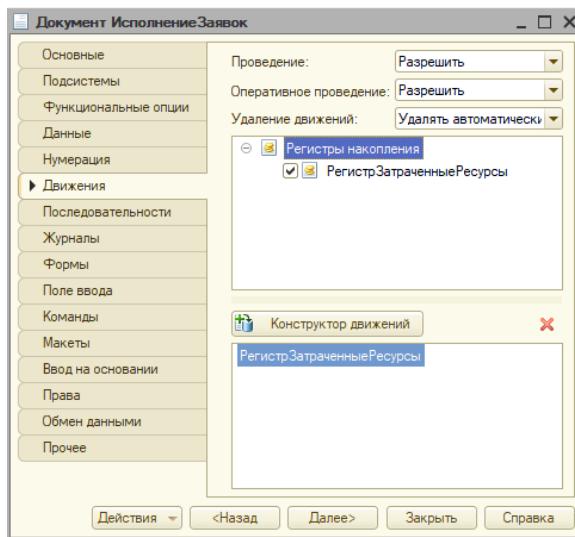
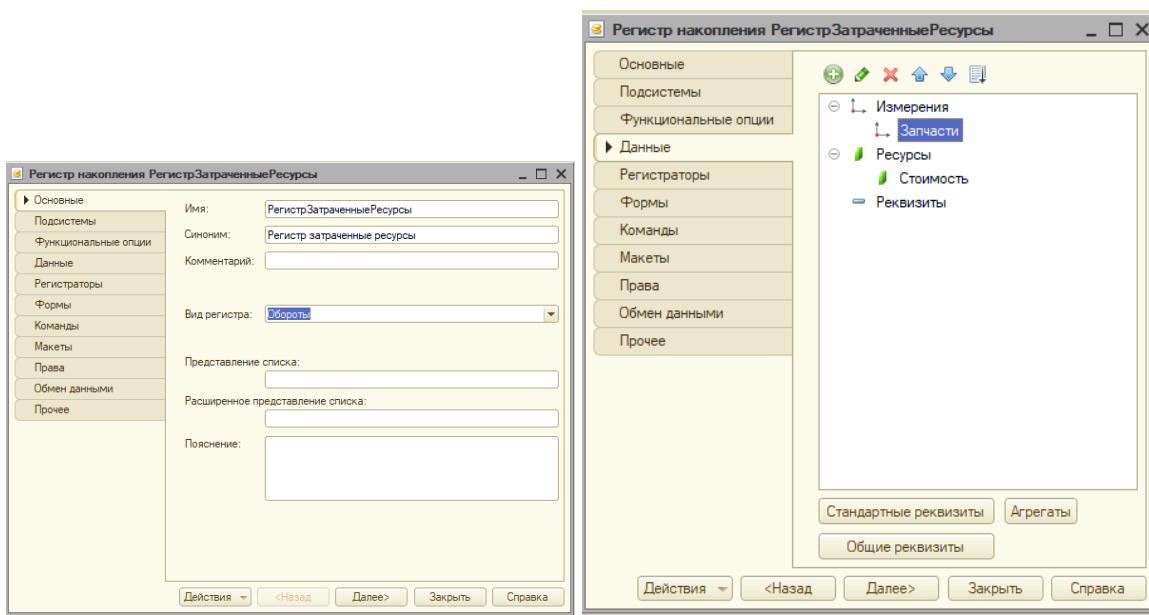


Напишем событие *ПриИзменении*

```
&НаКлиенте
Процедура ЗаказатьЗапчасти(Команда)
    ОткрытьФорму ("Документ.ЗаказЗапчастей.Форма.ФормаДокумента");
КонецПроцедуры
```

Запустите 1С: Предприятие и проверьте функционал кнопки и документа.

Отчётность и информирование: важной составляющей учёта заявок на ремонт является фиксация и отчёт о выполненной работе. После завершения ремонта, исполнитель должен предоставить отчёт о проделанной работе, включая информацию о затраченных ресурсах (время, материалы, стоимость), причине неисправности и оказанной помощи.



Добавьте в документ следующие

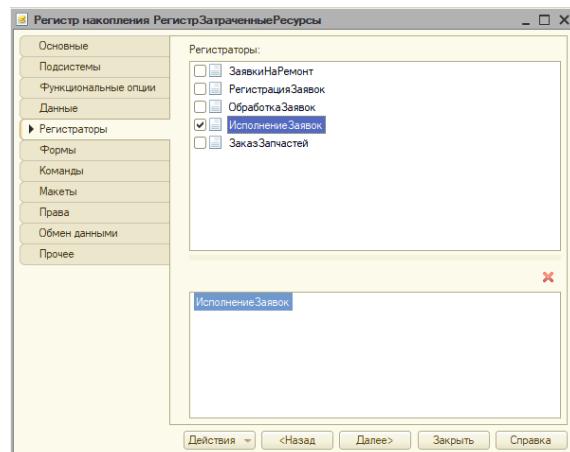
реквизиты:

**Запчасти** — Тип

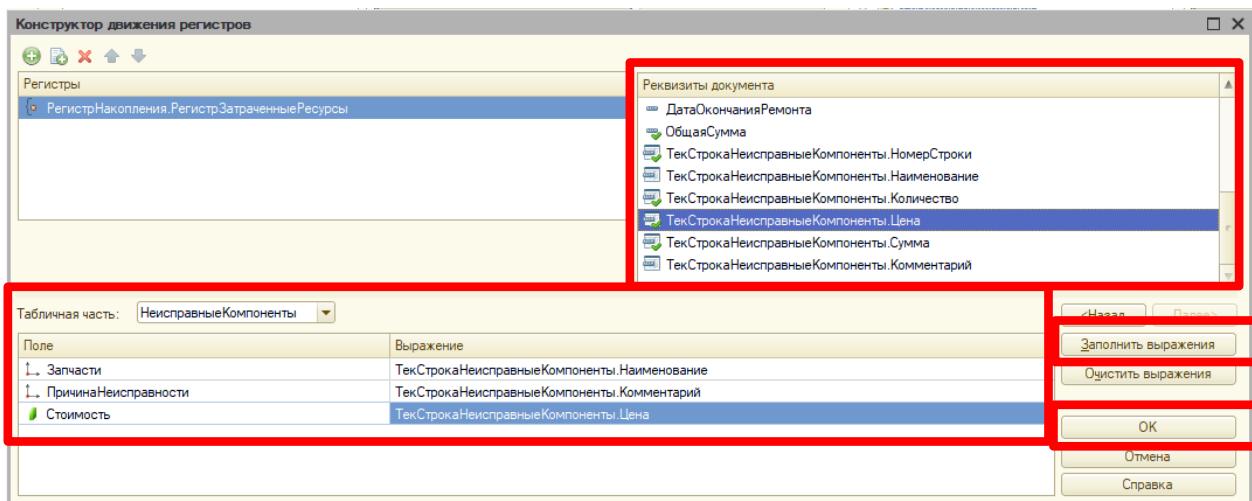
*СправочникСсылка.Запчасти;*

**Стоимость** – Тип Число, Длина 10, Точность 2,

*Неотрицательное;*



Теперь открываем документ **ИсполнениеЗаявки** и во вкладке Движение проводим следующие манипуляции. Создаем **Конструктор движения**.



В результате будет создана автоматически обработка

```

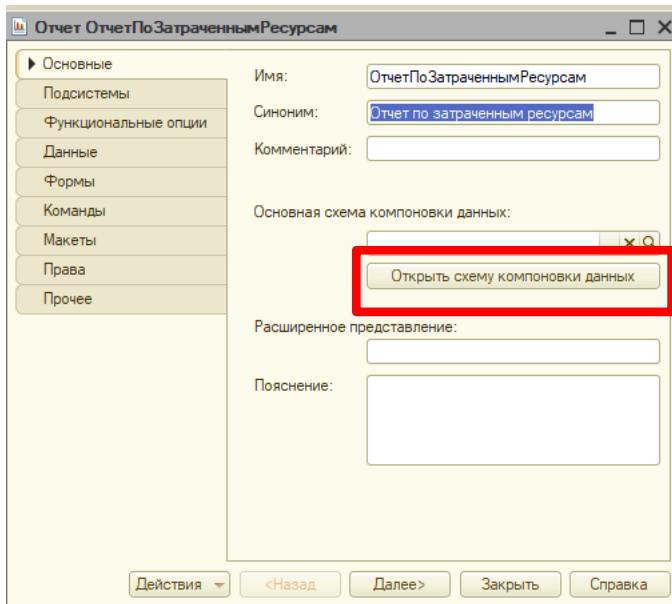
Процедура ОбработкаПроведения(Отказ, Режим)

// регистр РегистрЗатраченныеРесурсы
Движение.РегистрЗатраченныеРесурсы.Записывать = Истина;
Для Каждого ТекСтрокаНеисправныеКомпоненты Из НеисправныеКомпоненты Цикл
    Движение = Движение.РегистрЗатраченныеРесурсы.Добавить();
    Движение.Период = Дата;
    Движение.Запчасти = ТекСтрокаНеисправныеКомпоненты.Наименование;
    Движение.Стоимость = ТекСтрокаНеисправныеКомпоненты.Цена;
КонецЦикла;

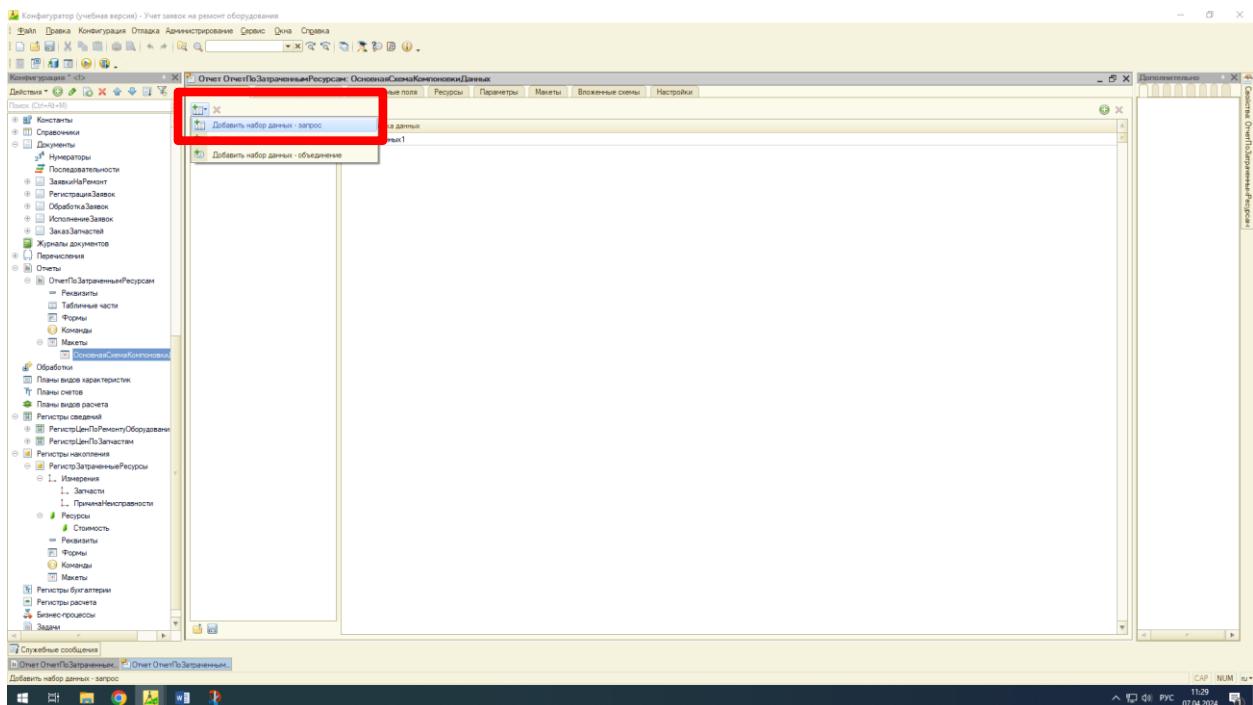
КонецПроцедуры

```

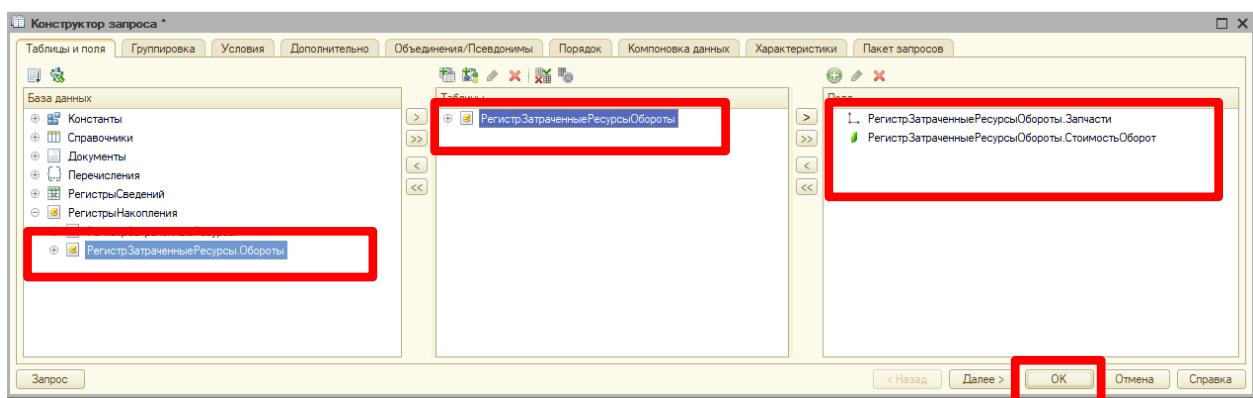
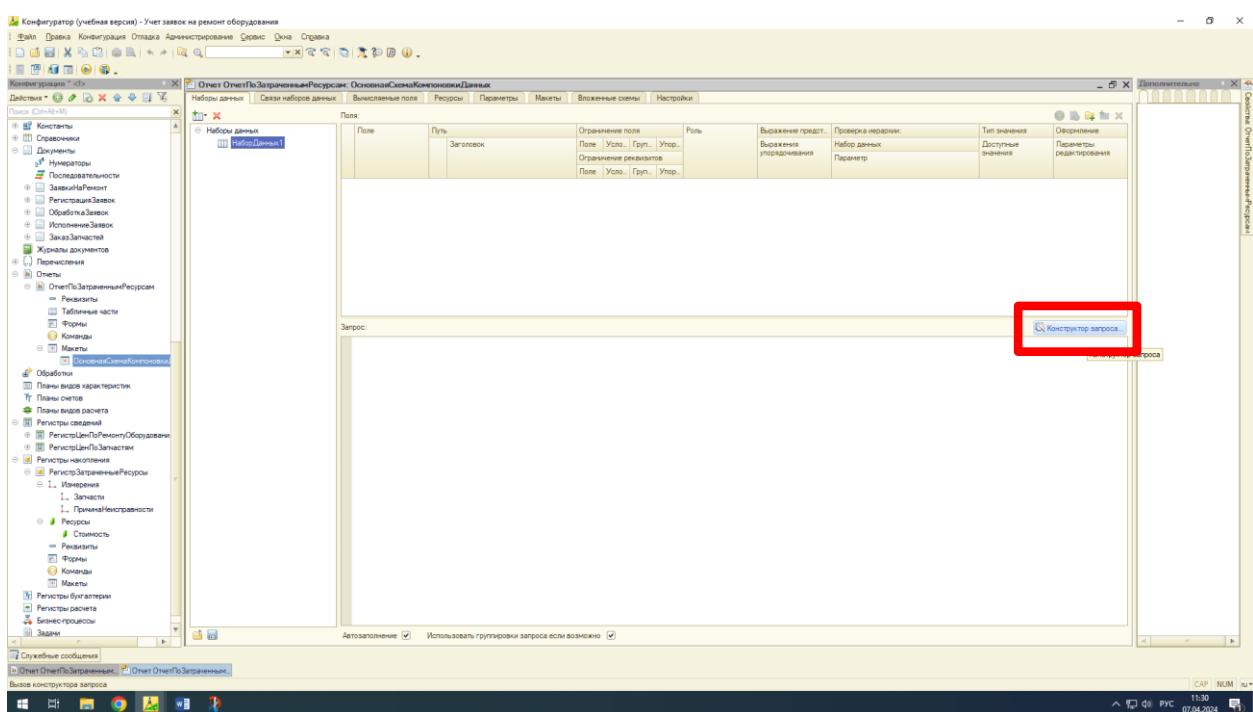
На основе данного регистра создаем Отчет по затраченным ресурсам, смотрим ниже:

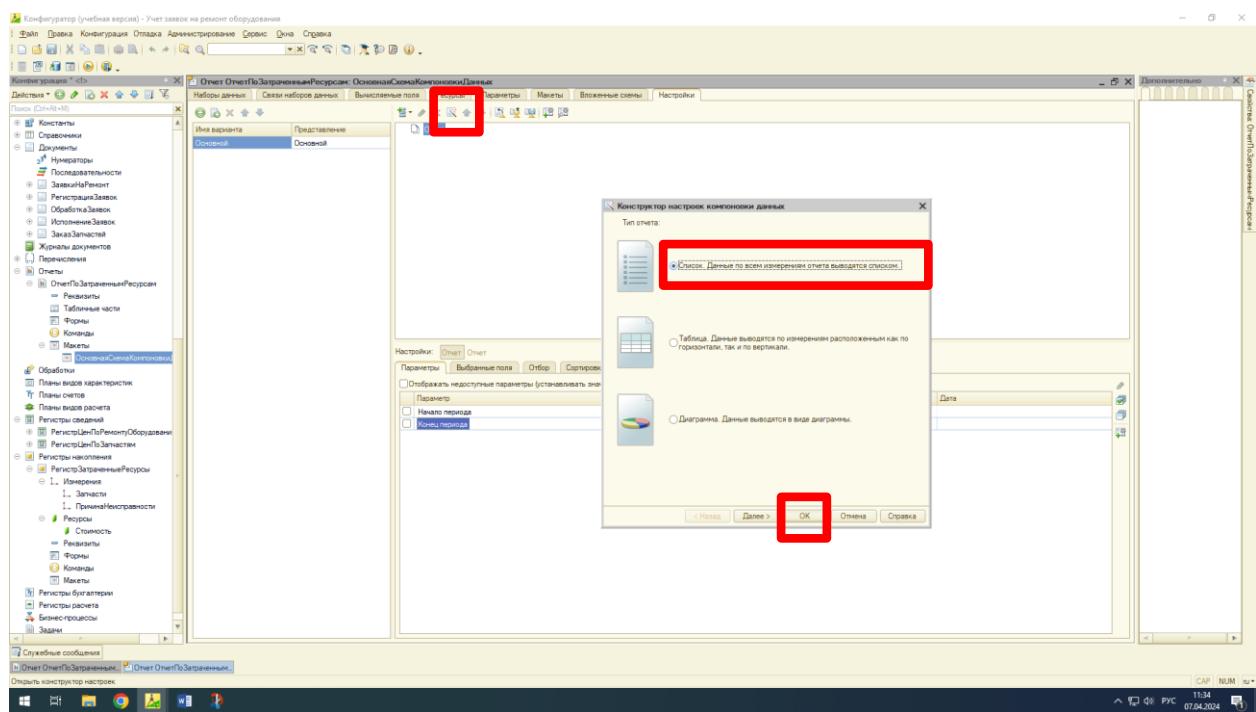
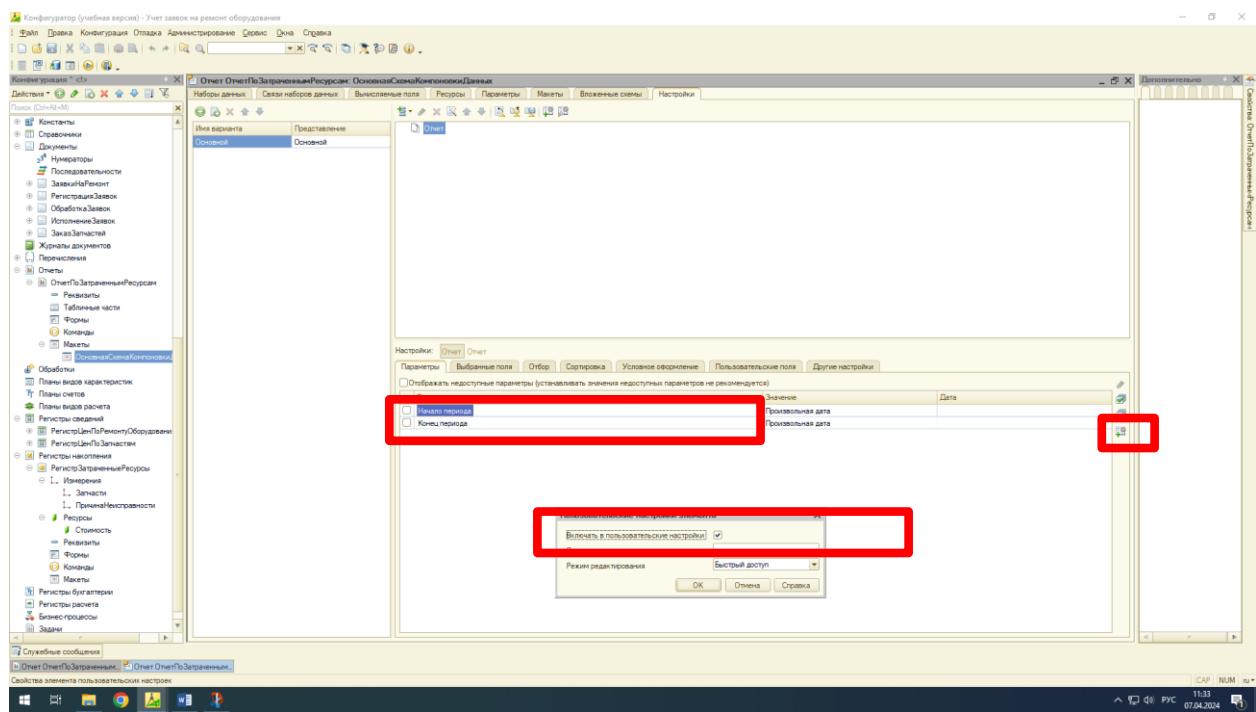


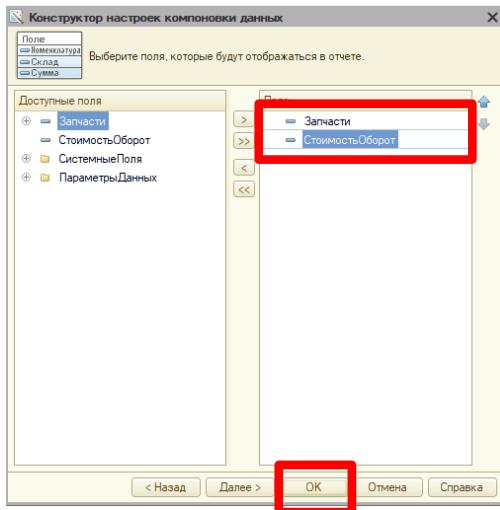
2 шаг



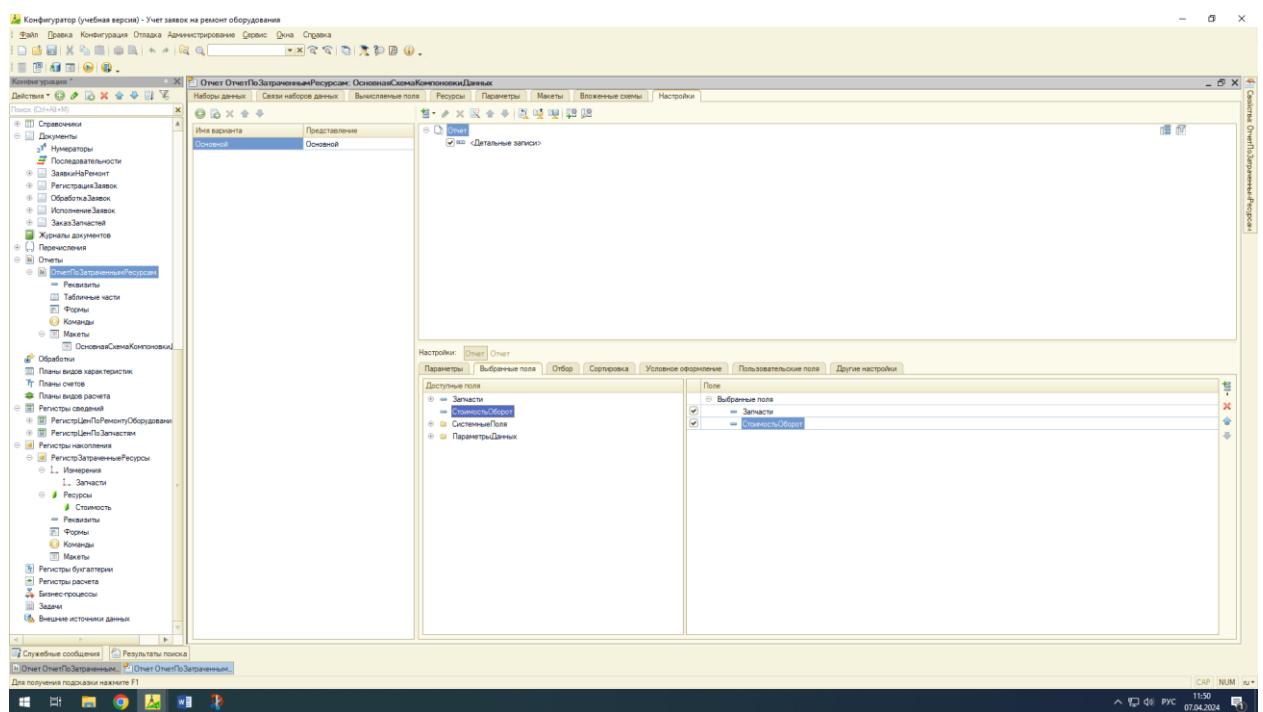
### 3 шаг

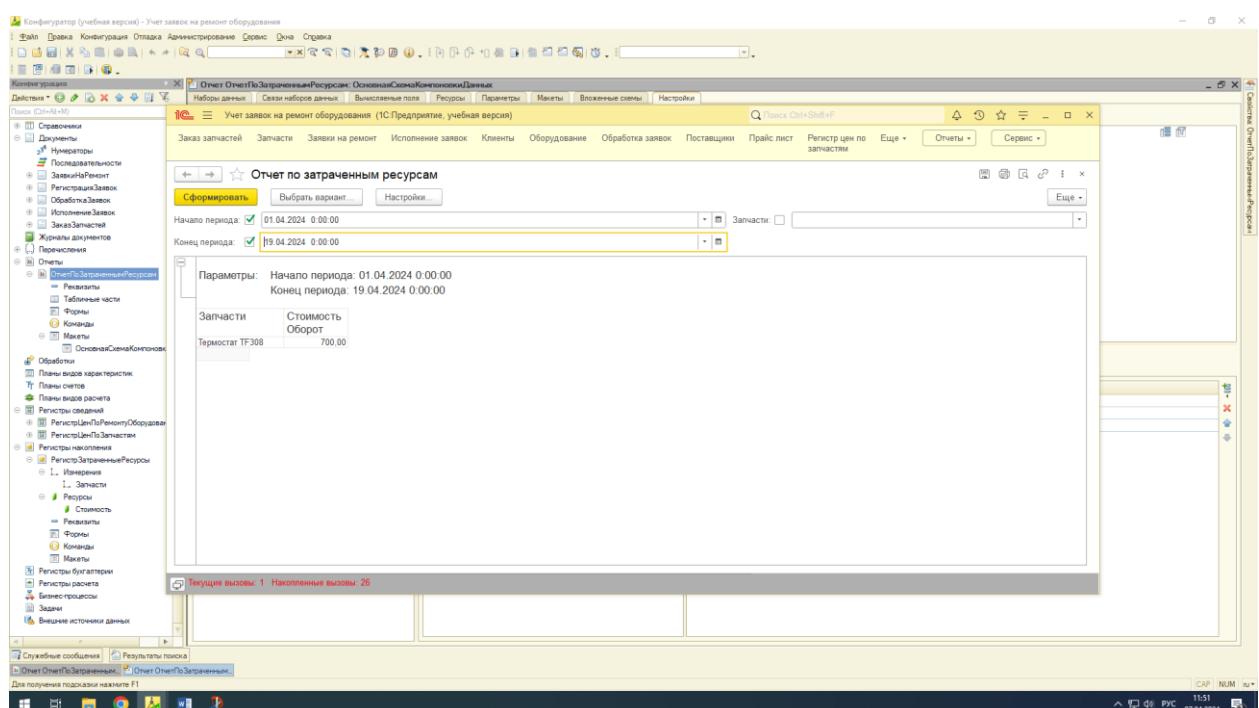
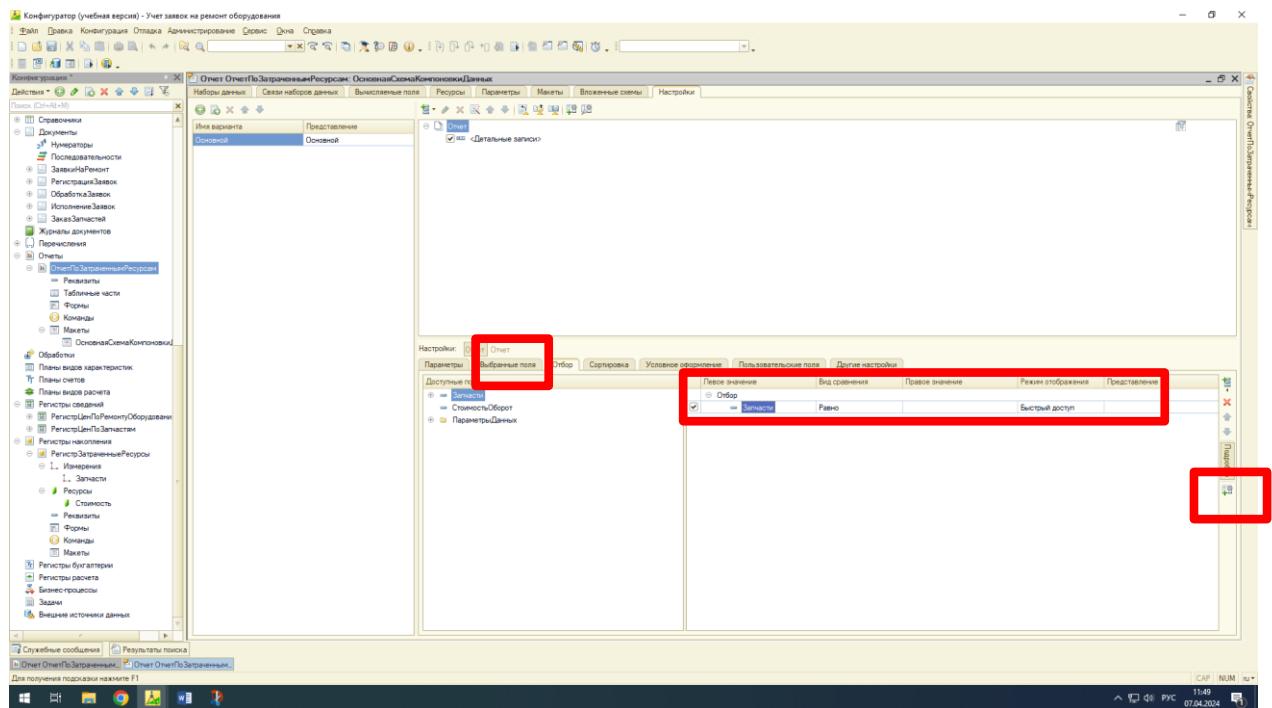




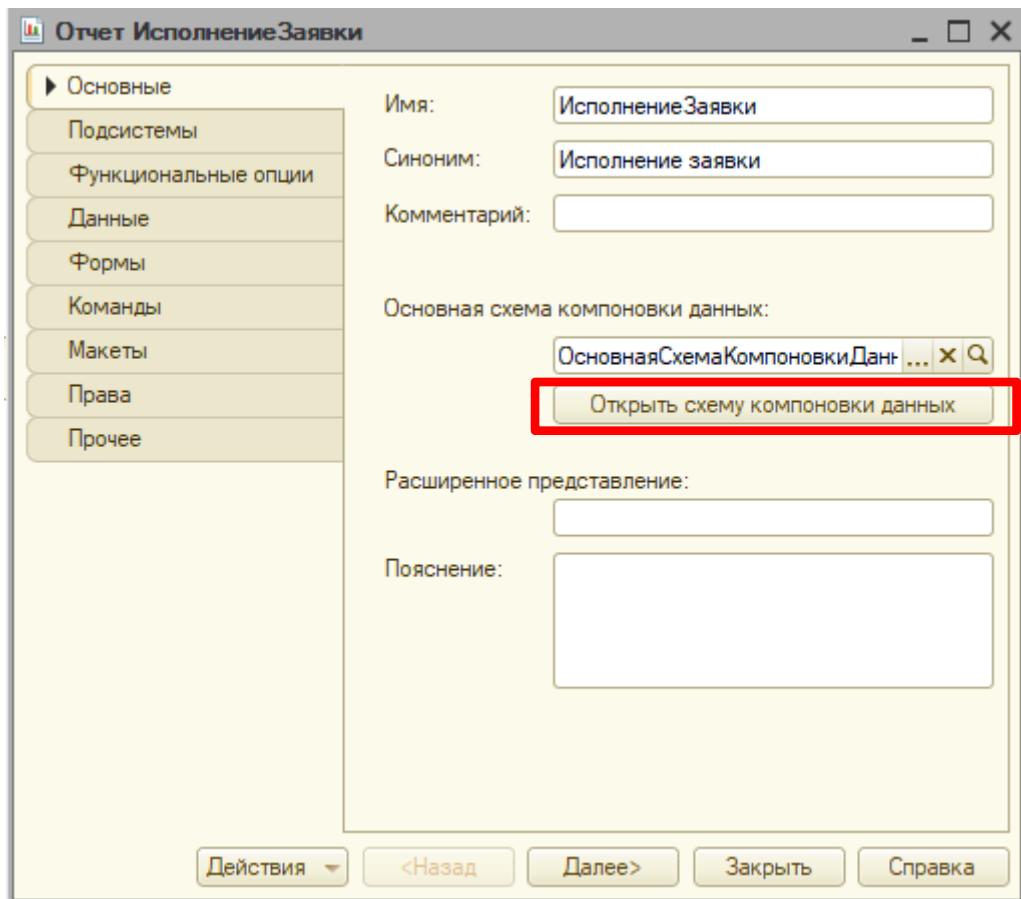


Далее без изменения и OK. Обеспечьте быстрый доступ.





По аналогии самостоятельно создайте отчет по документу ИсполнениеЗаявки на первую ТЧ, где выделите такие реквизиты как Оборудование, тип неисправности, цена.



**Отчет ИсполнениеЗаявки: ОсновнаяСхемаКомпоновкиДанных**

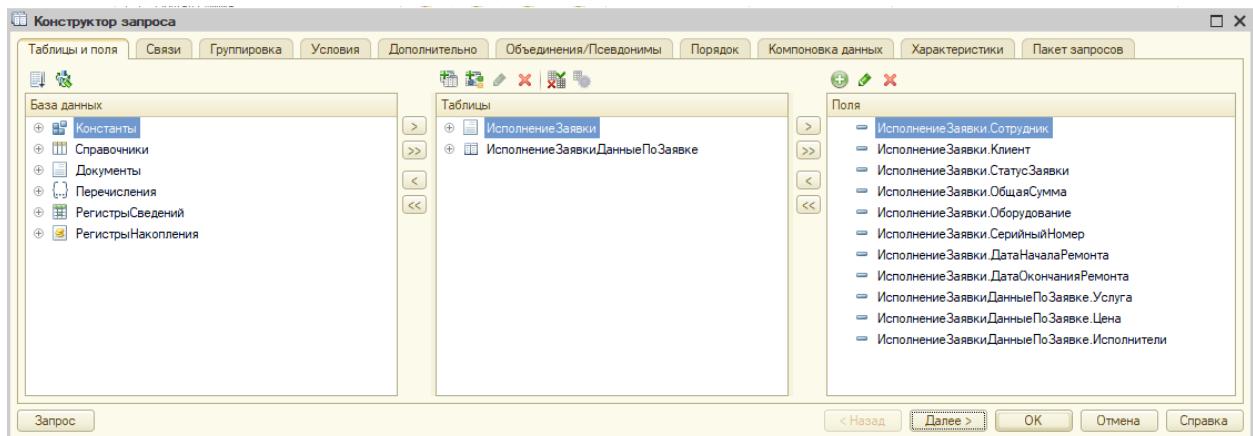
Наборы данных | Связи наборов данных | Вычисляемые поля | Ресурсы | Параметры | Макеты | Вложенные схемы | Настройки | Сервис

Поля

Поле	Путь	Ограничение поля	Роль	Выражение предст.	Проверка иерархи:	Тип значения	Об оформлени
	Заголовок	Поле   Усло..   Группа   Упор.. Ограничение реквизитов		Выражение упорядочивания	Набор данных Параметр	Доступные значения	Параметры редактирования
		Поле   Усло..   Группа   Упор..					

Запрос:

Автозаполнение  Использовать группировки запроса если возможно



Конфигуратор - Учет заявок на ремонт оборудования

Файл Домик Конфигуратор Отладка Архивирование Сервис Оана Справка

Отчет ИсполнениеЗаявки\_ОсновныеСведенияКонфигурацииДанных

Наборы данных Связи наборов данных Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки

Пол:

Наборы данных	Ресурсы	Выражение представлена	Проверка инерактивности	Набор данных	Доступные значения	Оформление
Любые	Автозаголовок	Пол., Услов., Группа, Упор., Ограничение реализации	Пол., Услов., Группа, Упор., Выражение упорядочивания	Параметр	Параметры редактирования	
Любые	ДатаНачалаРемонта	<input checked="" type="checkbox"/> Дата начала ремонта	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Любые	ОбщаяСумма	<input checked="" type="checkbox"/> Общая сумма	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Сотрудник	Сотрудник	<input checked="" type="checkbox"/> Сотрудник	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Клиент	Клиент	<input checked="" type="checkbox"/> Клиент	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
СтатусЗаявки	СтатусЗаявки	<input checked="" type="checkbox"/> Статус заявки	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Любые	ДатаОкончанияРемонта	<input checked="" type="checkbox"/> Дата окончания ремонта	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Любые	Оборудование	<input checked="" type="checkbox"/> Оборудование	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Любые	СерийныйНомер	<input checked="" type="checkbox"/> Серийный номер	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Услуга	Услуга	<input checked="" type="checkbox"/> Услуга	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			
Цена	Цена	<input checked="" type="checkbox"/> Цена	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>			

Запрос:

```

ВНЕРЯД
ИсполнениеЗаявки..Сотрудник КАК сотрудник,
ИсполнениеЗаявки..Клиент КАК клиент,
ИсполнениеЗаявки..СтатусЗаявки КАК статусЗаявки,
ИсполнениеЗаявки..ОбщаяСумма КАК общаяСумма,
ИсполнениеЗаявки..Оборудование КАК оборудование,
ИсполнениеЗаявки..СерийныйНомер КАК серийныйНомер,
ИсполнениеЗаявки..ДатаНачалаРемонта КАК датаНачалаРемонта,
ИсполнениеЗаявки..ДатаОкончанияРемонта КАК датаОкончанияРемонта,
ИсполнениеЗаявки..ДанныеПоЗаявке КАК данныеПоЗаявке,
ИсполнениеЗаявки..Услуга КАК услуга,
ИсполнениеЗаявки..Цена КАК цена,
ИсполнениеЗаявки..Исполнители КАК исполнители

```

ИЗ

```

Документ..ИсполнениеЗаявки..ДанныеПоЗаявке КАК ИсполнениеЗаявкиДанныеПоЗаявке
ЛЕВСО СОДИНЕННИЕ Документ..ИсполнениеЗаявки КАК ИсполнениеЗаявка
ДО ИсполнениеЗаявкиДанныеПоЗаявке..Ссылка КАК ИсполнениеЗаявки..Ссылка

```

Автозаполнение  Использовать группировки запроса если возможно

Отчет ИсполнениеЗаявки: ОсновнаяСхемаКомпоновкиДанных

Наборы данных | Связи наборов данных | Вычисляемые поля | Ресурсы | Параметры | Макеты | Вложенные схемы | Настройки

Имя варианта Представление  
Основной Основной

Отчет  
Сотрудник  
<Детальные записи>

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.

Настройки: Отчет Сотрудник  
Выбранные поля Отчет  
Доступные поля  
Исполнители Клиент Оборудование  
ОбщаяСумма СерийныйНомер  
Сотрудник ТипНеисправности  
СистемныеПоля

Сформировать Выбрать вариант... Настройки... Еще -

Дата начала ремонта	Дата окончания ремонта	Исполнители	Клиент	Оборудование	Общая сумма	Серийный номер	Сотрудник
01.05.2024	08.05.2024	Стреколовский Сергей Анатольевич	Петруня Петр Александрович	ПК	4 000.00	454КА41	Федоров Сергей Николаевич
01.05.2024	14.05.2024	Федоров Сергей Николаевич	Петруня Петр Александрович	ПК	4 000.00	445ОР14	Федоров Сергей Николаевич
08.05.2024	10.05.2024	Стреколовский Сергей Анатольевич	Петруня Петр Александрович	Пылесос Samsung VCC4520S3	4 500.00	кк4уее	Федоров Сергей Николаевич

Последнее что осталось сделать создать роли и назначит им определенный функционал в соответствии с должностью. Описание как это сделать тут <https://www.1s-up.ru/rol-1s/> Клиент может видеть только документ Заявка на ремонт и Регистр цен на типы неисправностей.