**БИЛЕТ 5**

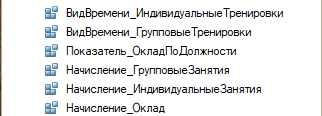
**РЕШЕНИЕ:**

**I КОНФИГУРАТОР.**

1. **Снимаем с поддержки, создаем подсистемы:**



1. **Делаем 6 констант (подсистема Экзамен-Прочее):**

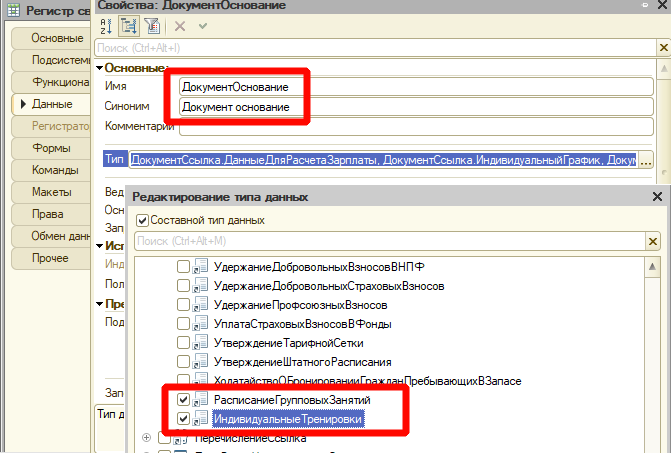


**ВидВремени\_ИндивидуальныеТренировки**, **ВидВремени\_ГрупповыеТренировки** - (СправочникСсылка.ВидыИспользованияРабочегоВремени),

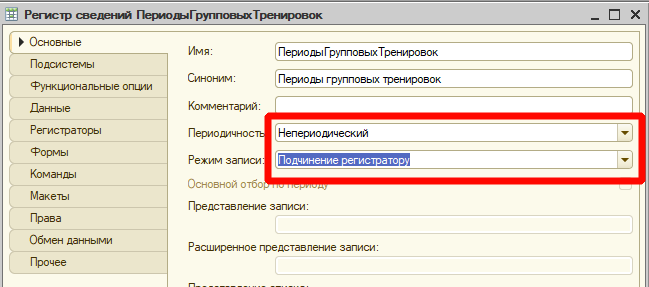
**Показатель\_ОкладПоДолжности** (СправочникСсылка.ПоказателиРасчетаЗарплаты),

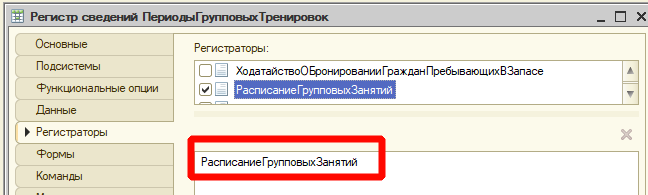
**Начисление\_ГрупповыеЗанятия**, **Начисление\_ИндивидуальныеЗанятия**, **Начисление\_Оклад** - ПланВидовРасчетаСсылка.Начисления.

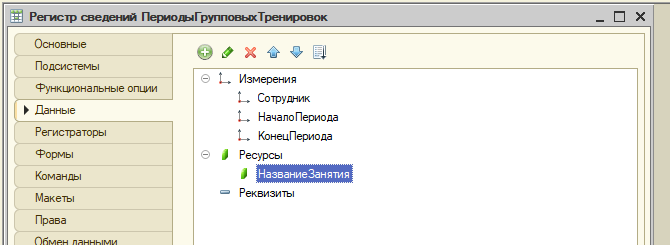
1. **Создаем два документа РасписаниеГрупповыхЗанятий и ИндивидуальныеТренировки (подсистема Экзамен-документы):**
2. **В регистре сведений ПараметрыЗарегистрированныхДанныхУчетаВремениСотрудников – в измерении ДокументОснование – добавляем в состав ДВА вновь добавленных документа:**



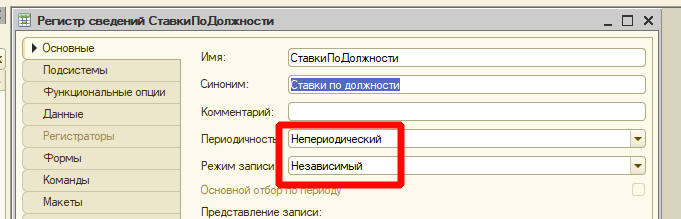
1. **Добавляем два регистра сведений ПериодыГрупповыхТренировок и СтавкиПоДолжности (подсистема-экзамен-прочее):**

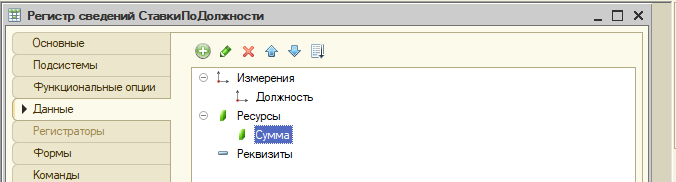




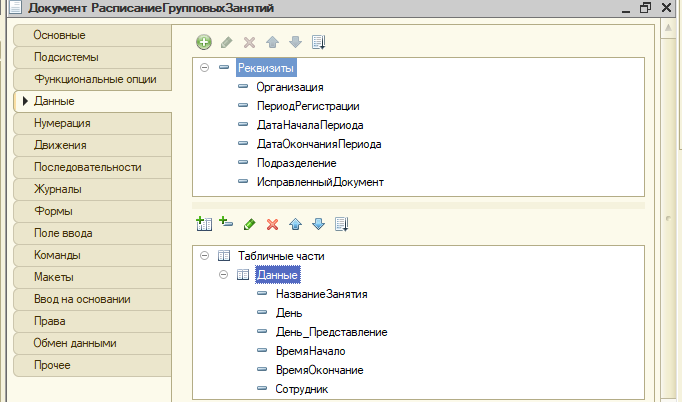


Измерения (3): **Сотрудник** (СправочникСсылка.Сотрудники), **НачалоПериода** и **КонецПериода** – (Дата , **состав** Дата и Время) Ресурсы (1) : **НазваниеЗанятия** (строка 300)



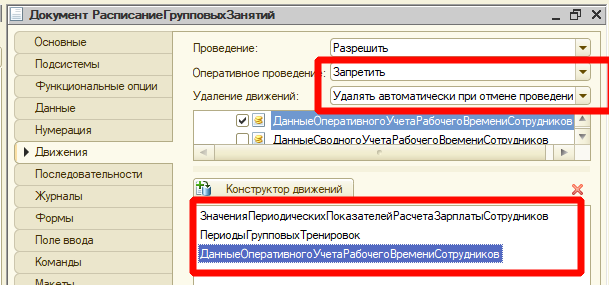


Измерения(1): **Должность** (СправочникСсылка.Должности), Ресурсы: **Сумма** (Число 10.2)

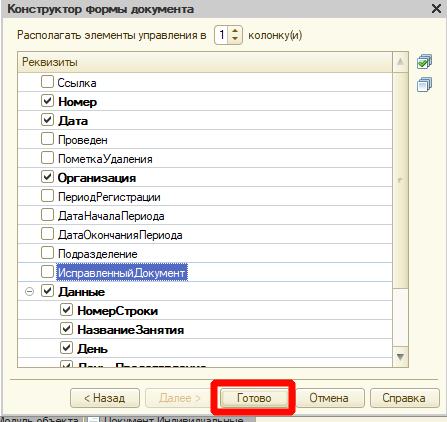
1. **Работаем с документом РасписаниеГрупповыхЗанятий.** 

**Шапка**: **Организация** (СправочникСсылка.Организации), **ПериодРегистрации** , **ДатаНачалаПериода**, **ДатаОкончанияПериода** – Дата, **Подразделение** - СправочникСсылка.ПодразделенияОрганизаций, **ИсправленныйДокумент** – Булево.

Табл.часть: **Данные** : **НазваниеЗанятия** (строка 300), **День** (Дата), **День\_Представление** (строка 50), **ВремяНачало** и **ВремяОкончание** (Дата, состав Время!!!), **Сотрудник** (СправочникСсылка.Сотрудники)



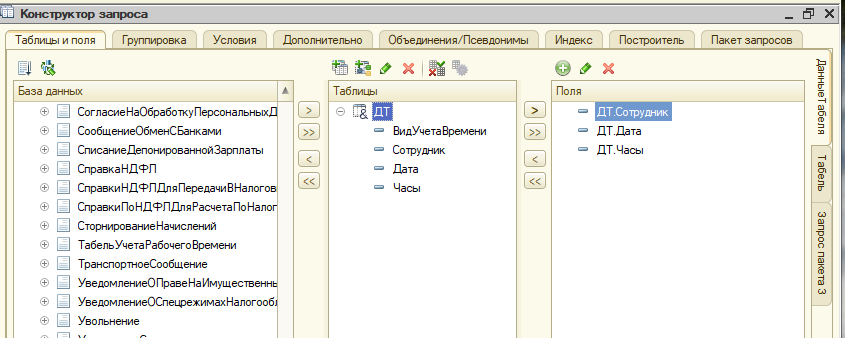
1. **Создаем форму:**



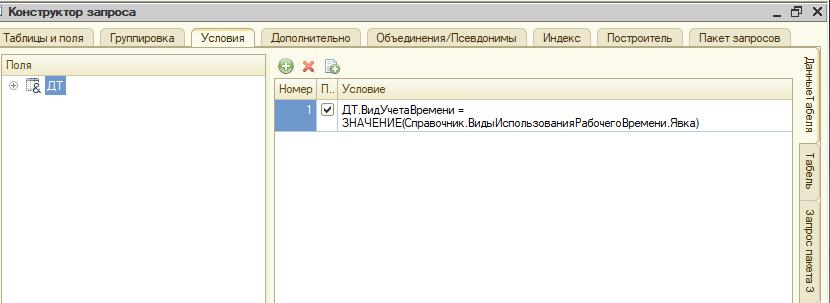
1. **Модуль документа РасписаниеГрупповыхЗанятий:**

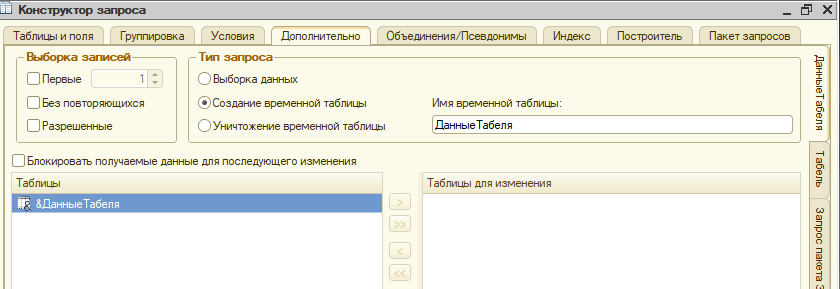
**ЗАПРОС:**

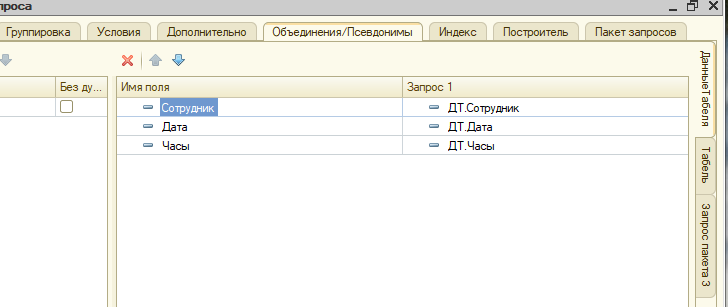
**ПЕРВЫЙ ПАКЕТ:**



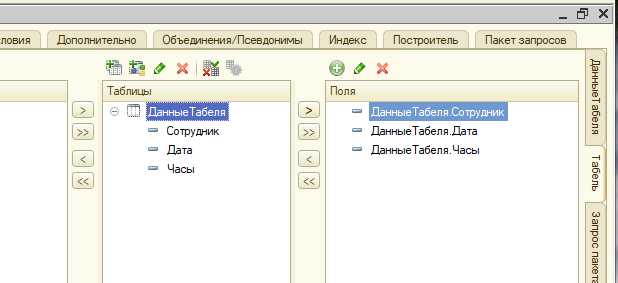
**Если не удается вписать условие, делаем вручную потом:**

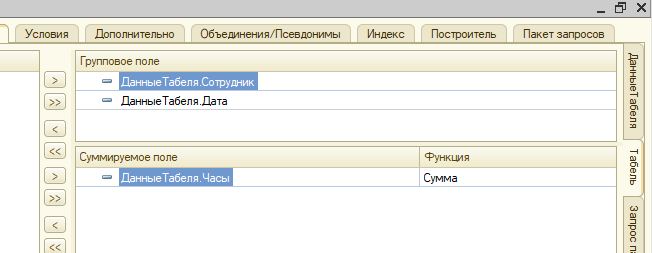


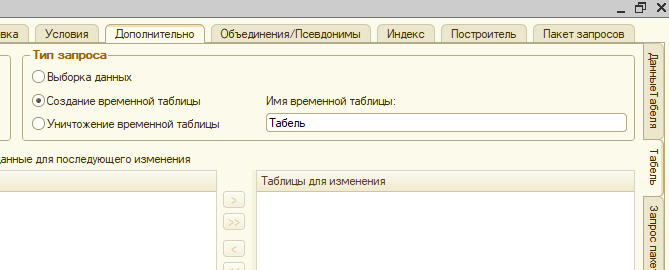


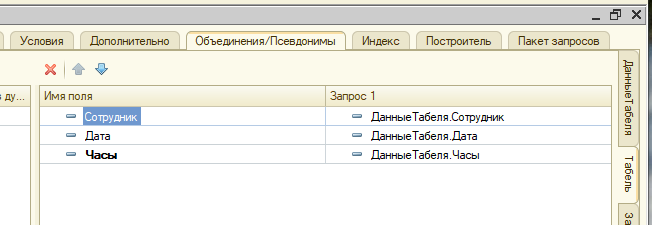


**ВТОРОЙ ПАКЕТ:**



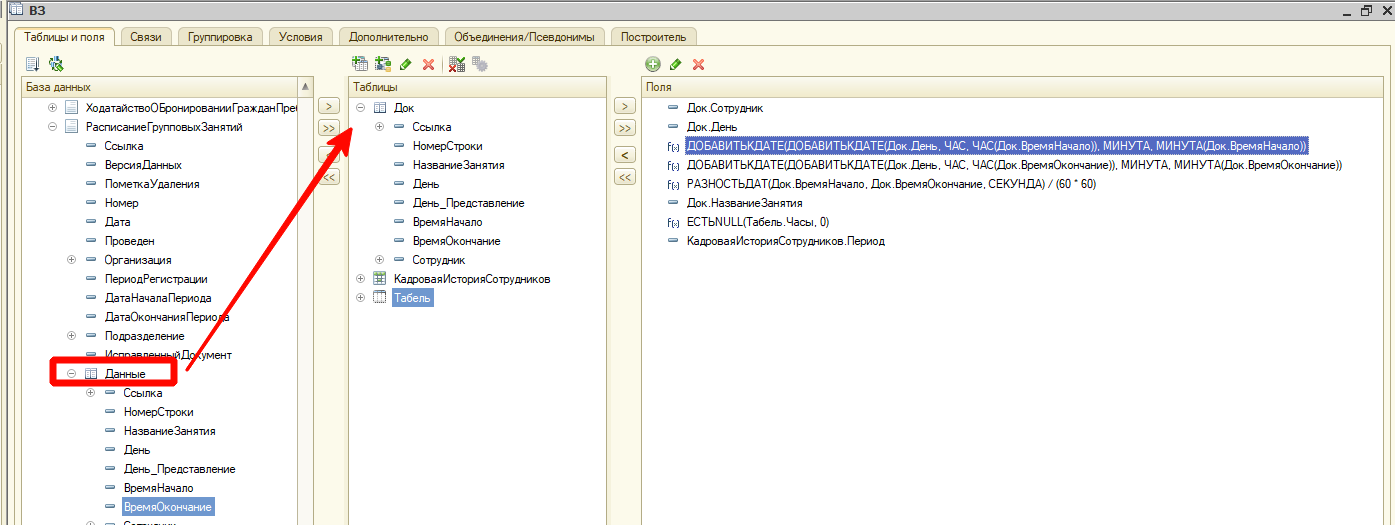


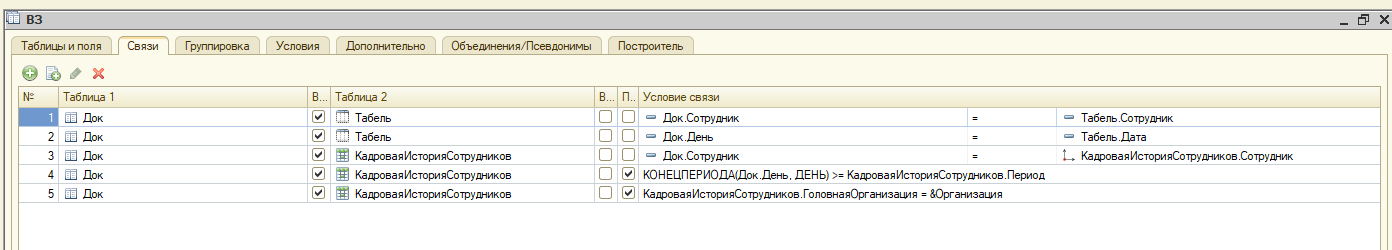


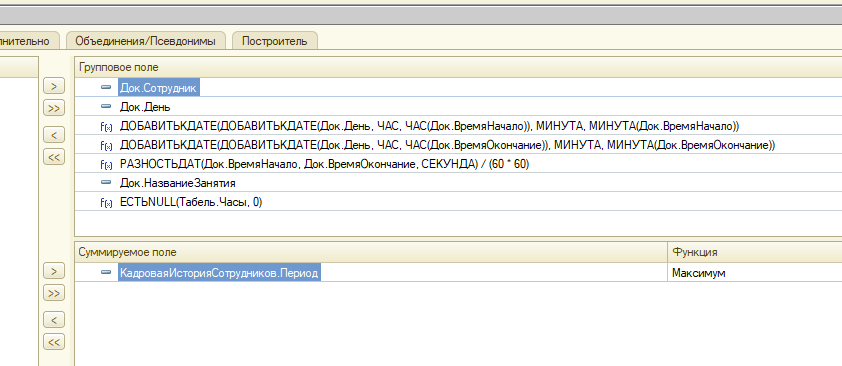


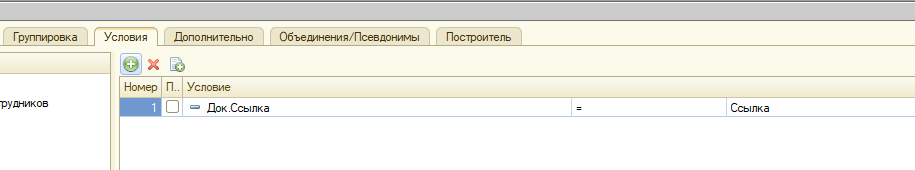
**ТРЕТИЙ ПАКЕТ:**

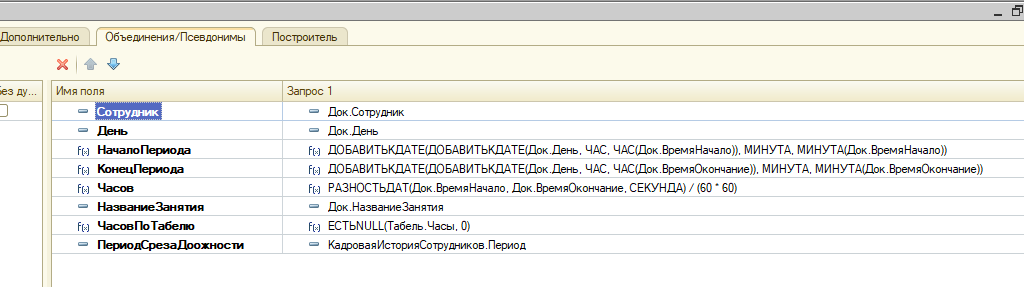
**НАЧАЛО ВЛОЖЕННОГО ЗАПРОСА ВЗ:**



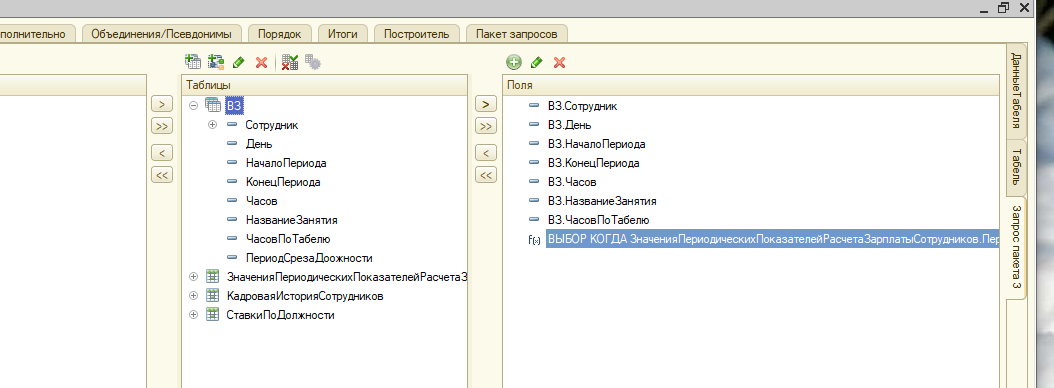


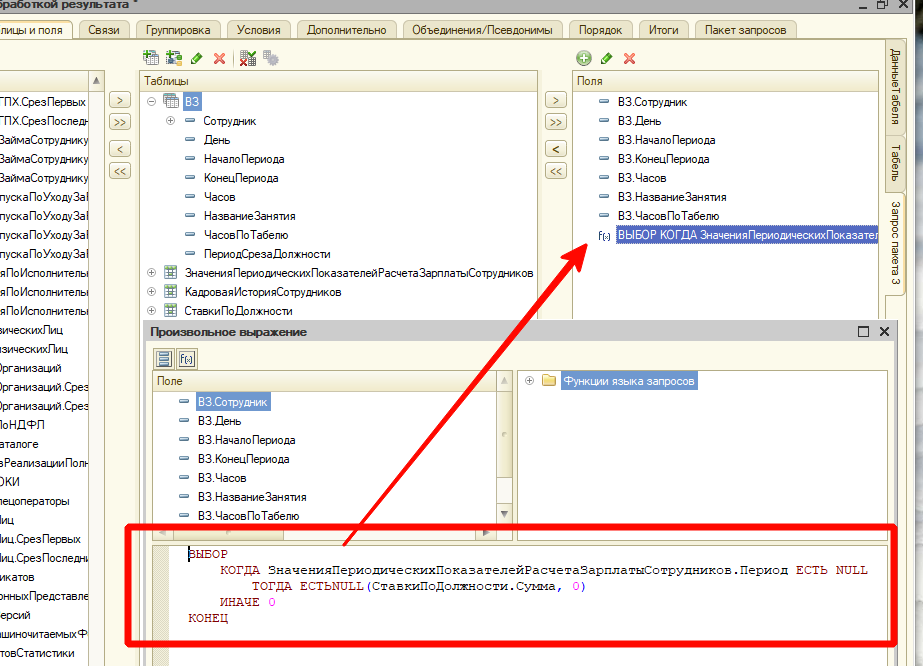


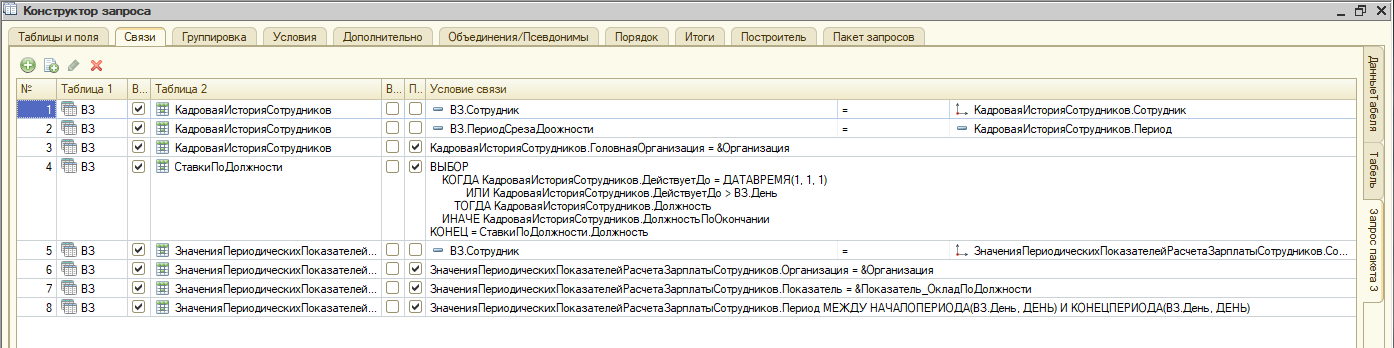


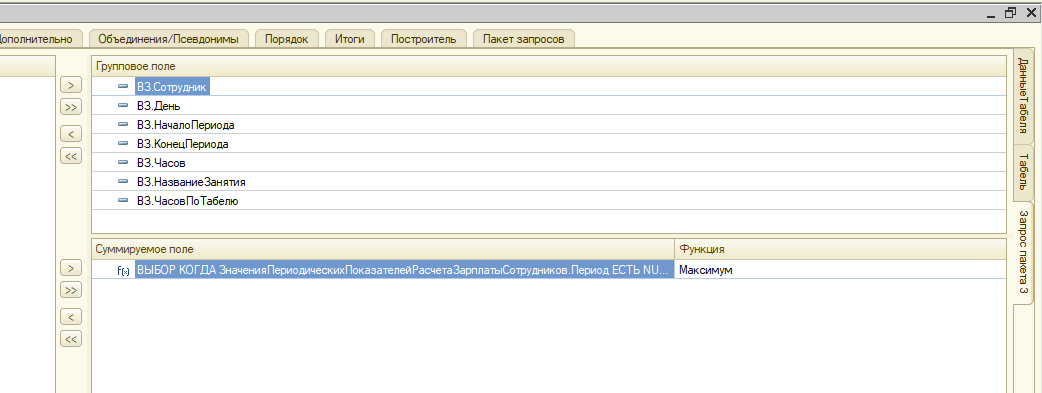


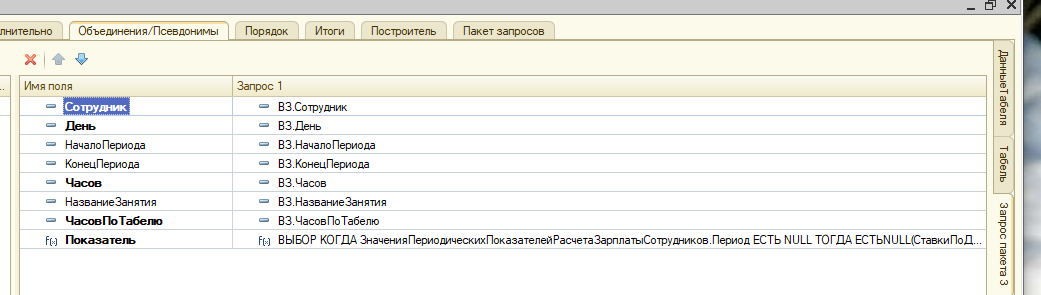
**ОКОНЧАНИЕ ВЛОЖЕННОГО ЗАПРОСА ВЗ**

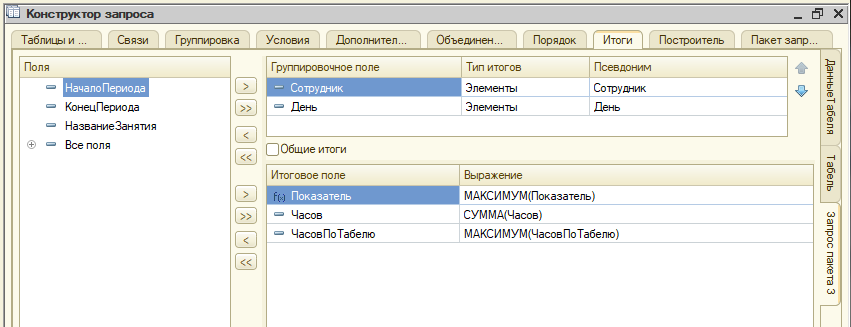


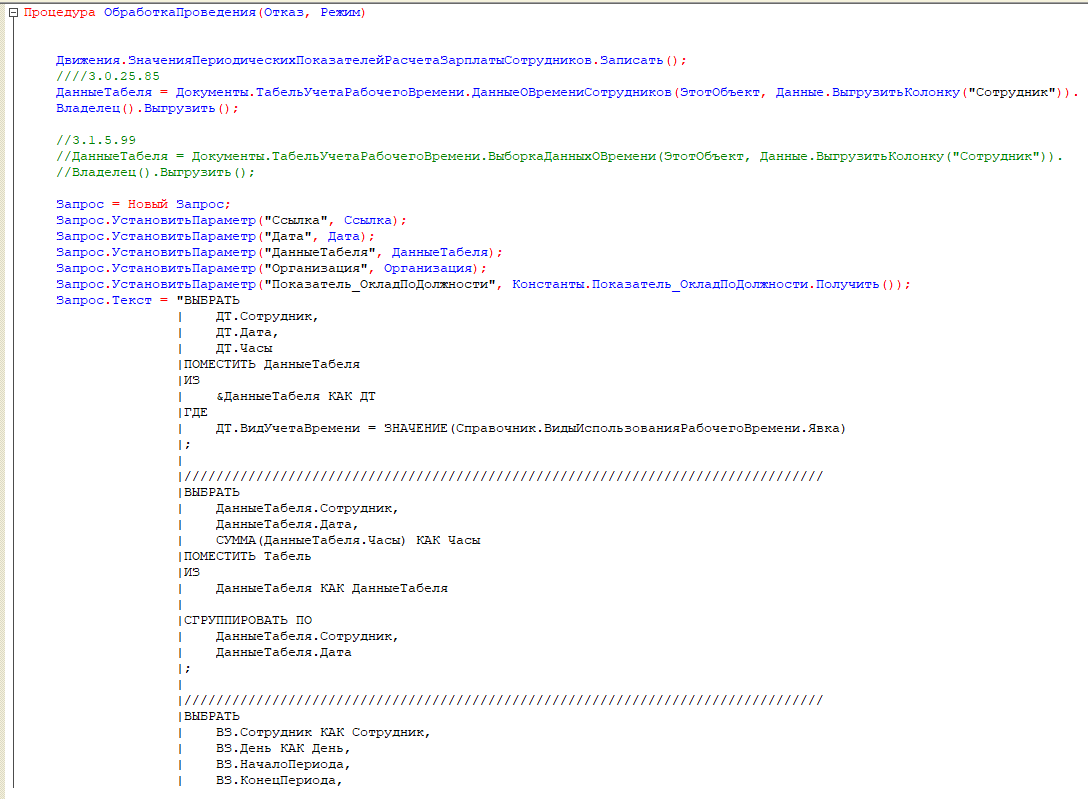


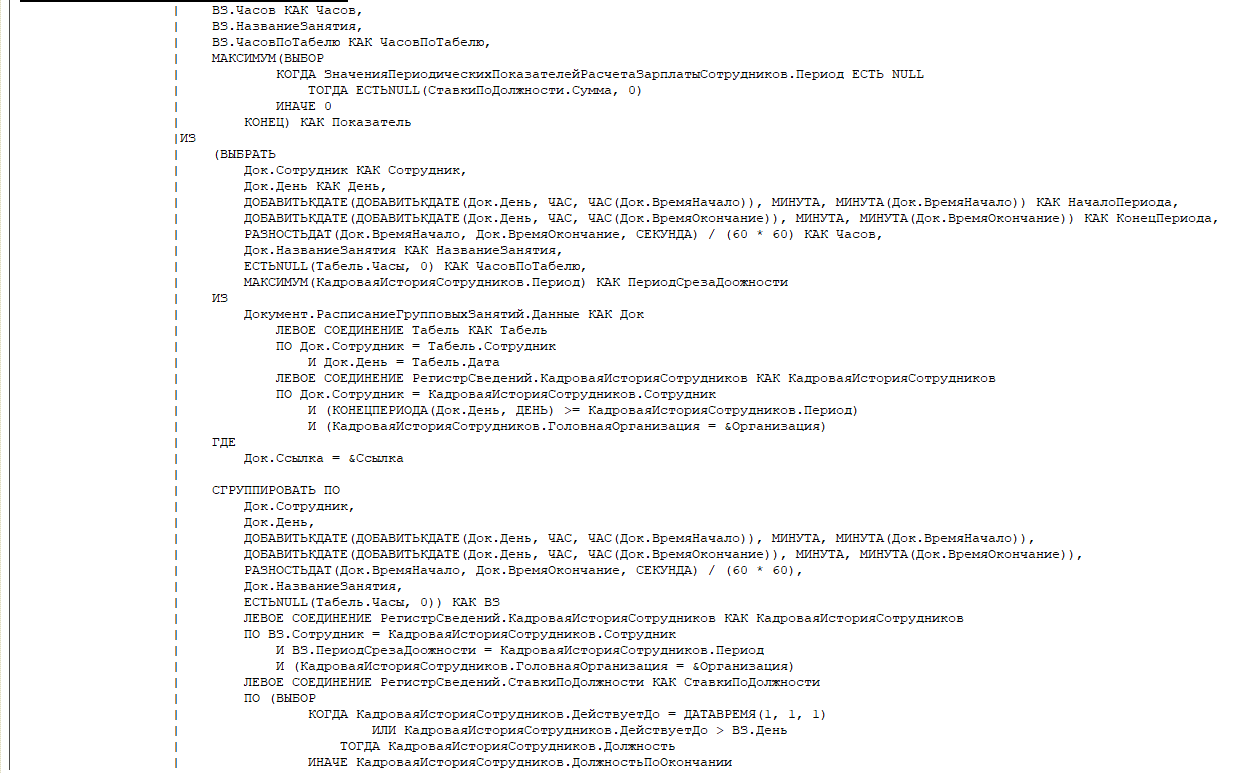






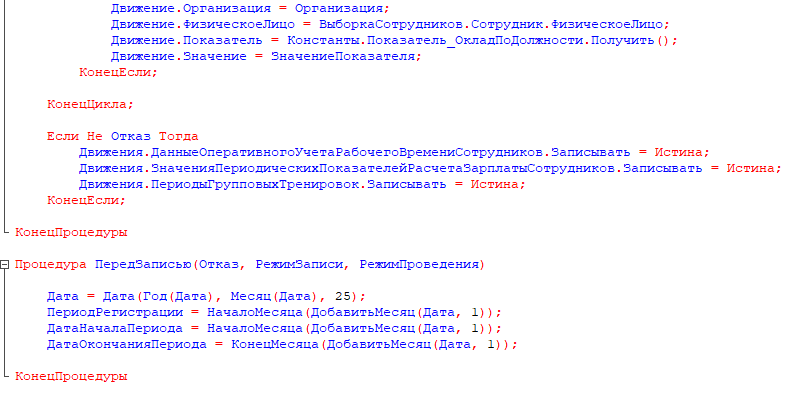












Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Записать();

////3.0.25.85

ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ДанныеОВремениСотрудников(ЭтотОбъект, Данные.ВыгрузитьКолонку("Сотрудник")).

Владелец().Выгрузить();

//3.1.5.99

//ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ВыборкаДанныхОВремени(ЭтотОбъект, Данные.ВыгрузитьКолонку("Сотрудник")).

//Владелец().Выгрузить();

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("ДанныеТабеля", ДанныеТабеля);

Запрос.УстановитьПараметр("Организация", Организация);

Запрос.УстановитьПараметр("Показатель\_ОкладПоДолжности", Константы.Показатель\_ОкладПоДолжности.Получить());

Запрос.Текст = "ВЫБРАТЬ

| ДТ.Сотрудник,

| ДТ.Дата,

| ДТ.Часы

|ПОМЕСТИТЬ ДанныеТабеля

|ИЗ

| &ДанныеТабеля КАК ДТ

|ГДЕ

| ДТ.ВидУчетаВремени = ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата,

| СУММА(ДанныеТабеля.Часы) КАК Часы

|ПОМЕСТИТЬ Табель

|ИЗ

| ДанныеТабеля КАК ДанныеТабеля

|

|СГРУППИРОВАТЬ ПО

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ВЗ.Сотрудник КАК Сотрудник,

| ВЗ.День КАК День,

| ВЗ.НачалоПериода,

| ВЗ.КонецПериода,

| ВЗ.Часов КАК Часов,

| ВЗ.НазваниеЗанятия,

| ВЗ.ЧасовПоТабелю КАК ЧасовПоТабелю,

| МАКСИМУМ(ВЫБОР

| КОГДА ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Период ЕСТЬ NULL

| ТОГДА ЕСТЬNULL(СтавкиПоДолжности.Сумма, 0)

| ИНАЧЕ 0

| КОНЕЦ) КАК Показатель

|ИЗ

| (ВЫБРАТЬ

| Док.Сотрудник КАК Сотрудник,

| Док.День КАК День,

| ДОБАВИТЬКДАТЕ(ДОБАВИТЬКДАТЕ(Док.День, ЧАС, ЧАС(Док.ВремяНачало)), МИНУТА, МИНУТА(Док.ВремяНачало)) КАК НачалоПериода,

| ДОБАВИТЬКДАТЕ(ДОБАВИТЬКДАТЕ(Док.День, ЧАС, ЧАС(Док.ВремяОкончание)), МИНУТА, МИНУТА(Док.ВремяОкончание)) КАК КонецПериода,

| РАЗНОСТЬДАТ(Док.ВремяНачало, Док.ВремяОкончание, СЕКУНДА) / (60 \* 60) КАК Часов,

| Док.НазваниеЗанятия КАК НазваниеЗанятия,

| ЕСТЬNULL(Табель.Часы, 0) КАК ЧасовПоТабелю,

| МАКСИМУМ(КадроваяИсторияСотрудников.Период) КАК ПериодСрезаДоожности

| ИЗ

| Документ.РасписаниеГрупповыхЗанятий.Данные КАК Док

| ЛЕВОЕ СОЕДИНЕНИЕ Табель КАК Табель

| ПО Док.Сотрудник = Табель.Сотрудник

| И Док.День = Табель.Дата

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КадроваяИсторияСотрудников КАК КадроваяИсторияСотрудников

| ПО Док.Сотрудник = КадроваяИсторияСотрудников.Сотрудник

| И (КОНЕЦПЕРИОДА(Док.День, ДЕНЬ) >= КадроваяИсторияСотрудников.Период)

| И (КадроваяИсторияСотрудников.ГоловнаяОрганизация = &Организация)

| ГДЕ

| Док.Ссылка = &Ссылка

|

| СГРУППИРОВАТЬ ПО

| Док.Сотрудник,

| Док.День,

| ДОБАВИТЬКДАТЕ(ДОБАВИТЬКДАТЕ(Док.День, ЧАС, ЧАС(Док.ВремяНачало)), МИНУТА, МИНУТА(Док.ВремяНачало)),

| ДОБАВИТЬКДАТЕ(ДОБАВИТЬКДАТЕ(Док.День, ЧАС, ЧАС(Док.ВремяОкончание)), МИНУТА, МИНУТА(Док.ВремяОкончание)),

| РАЗНОСТЬДАТ(Док.ВремяНачало, Док.ВремяОкончание, СЕКУНДА) / (60 \* 60),

| Док.НазваниеЗанятия,

| ЕСТЬNULL(Табель.Часы, 0)) КАК ВЗ

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КадроваяИсторияСотрудников КАК КадроваяИсторияСотрудников

| ПО ВЗ.Сотрудник = КадроваяИсторияСотрудников.Сотрудник

| И ВЗ.ПериодСрезаДоожности = КадроваяИсторияСотрудников.Период

| И (КадроваяИсторияСотрудников.ГоловнаяОрганизация = &Организация)

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.СтавкиПоДолжности КАК СтавкиПоДолжности

| ПО (ВЫБОР

| КОГДА КадроваяИсторияСотрудников.ДействуетДо = ДАТАВРЕМЯ(1, 1, 1)

| ИЛИ КадроваяИсторияСотрудников.ДействуетДо > ВЗ.День

| ТОГДА КадроваяИсторияСотрудников.Должность

| ИНАЧЕ КадроваяИсторияСотрудников.ДолжностьПоОкончании

// | ИНАЧЕ КадроваяИсторияСотрудников.УдалитьДолжностьПоОкончании // так для 3.1

| КОНЕЦ = СтавкиПоДолжности.Должность)

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников КАК ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников

| ПО ВЗ.Сотрудник = ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Сотрудник

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Организация = &Организация)

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Показатель = &Показатель\_ОкладПоДолжности)

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Период МЕЖДУ НАЧАЛОПЕРИОДА(ВЗ.День, ДЕНЬ) И КОНЕЦПЕРИОДА(ВЗ.День, ДЕНЬ))

|

|СГРУППИРОВАТЬ ПО

| ВЗ.Сотрудник,

| ВЗ.День,

| ВЗ.НачалоПериода,

| ВЗ.КонецПериода,

| ВЗ.Часов,

| ВЗ.НазваниеЗанятия,

| ВЗ.ЧасовПоТабелю

|ИТОГИ

| СУММА(Часов),

| МАКСИМУМ(ЧасовПоТабелю),

| МАКСИМУМ(Показатель)

|ПО

| Сотрудник,

| День

|";

ВыборкаСотрудников= Запрос.Выполнить().Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Пока ВыборкаСотрудников.Следующий() Цикл

ЗначениеПоказателя = 0;

НачалоПериода = Неопределено;

ВыборкаДень = ВыборкаСотрудников.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Пока ВыборкаДень.Следующий() Цикл

Если ВыборкаДень.ЧасовПоТабелю < ВыборкаДень.Часов Тогда

СообщениеПользвателю = Новый СообщениеПользователю;

СообщениеПользвателю.Текст = "Сотрудник " + Строка(ВыборкаСотрудников.Сотрудник) + " " +

Формат(ВыборкаДень.День, "ДЛФ=DD") + " работает всего " + Строка(ВыборкаДень.ЧасовПоТабелю);

СообщениеПользвателю.Сообщить();

Отказ = Истина;

КонецЕсли;

Если Отказ Тогда

Продолжить;

КонецЕсли;

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = ВыборкаДень.День;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(ВыборкаДень.День);

Движение.ВидУчетаВремени = Константы.ВидВремени\_ГрупповыеТренировки.Получить();

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДополнительноеВнутрисменноеВремя;

Движение.Часы = ВыборкаДень.Часов;

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = ВыборкаДень.День;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(ВыборкаДень.День);

Движение.ВидУчетаВремени = Справочники.ВидыИспользованияРабочегоВремени.Явка;

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДанныеВнутрисменныхНеявок;

Движение.Часы = ВыборкаДень.Часов \* -1;

Выборка = ВыборкаДень.Выбрать();

Пока Выборка.Следующий() Цикл

Движение = Движения.ПериодыГрупповыхТренировок.Добавить();

ЗаполнитьЗначенияСвойств(Движение, Выборка);

КонецЦикла;

Если ЗначениеПоказателя = 0 Тогда

ЗначениеПоказателя = ВыборкаДень.Показатель;

НачалоПериода = ВыборкаДень.День;

Продолжить;

КонецЕсли;

Если ЗначениеПоказателя <> ВыборкаДень.Показатель И Не ЗначениеПоказателя = 0 Тогда

Движение = Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Добавить();

Движение.Период = НачалоПериода;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.Организация = Организация;

Движение.ФизическоеЛицо = ВыборкаСотрудников.Сотрудник.ФизическоеЛицо;

Движение.Показатель = Константы.Показатель\_ОкладПоДолжности.Получить();

Движение.Значение = ЗначениеПоказателя;

Движение.ДействуетДо = ВыборкаДень.День;

ЗначениеПоказателя = ВыборкаДень.Показатель;

НачалоПериода = ВыборкаДень.День;

Продолжить;

КонецЕсли;

КонецЦикла;

Если Не ЗначениеПоказателя = 0 Тогда

Движение = Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Добавить();

Движение.Период = НачалоПериода;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.Организация = Организация;

Движение.ФизическоеЛицо = ВыборкаСотрудников.Сотрудник.ФизическоеЛицо;

Движение.Показатель = Константы.Показатель\_ОкладПоДолжности.Получить();

Движение.Значение = ЗначениеПоказателя;

КонецЕсли;

КонецЦикла;

Если Не Отказ Тогда

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Записывать = Истина;

Движения.ПериодыГрупповыхТренировок.Записывать = Истина;

КонецЕсли;

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Дата = Дата(Год(Дата), Месяц(Дата), 25);

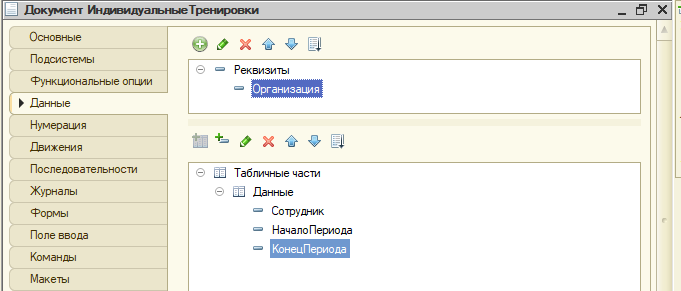
ПериодРегистрации = НачалоМесяца(ДобавитьМесяц(Дата, 1));

ДатаНачалаПериода = НачалоМесяца(ДобавитьМесяц(Дата, 1));

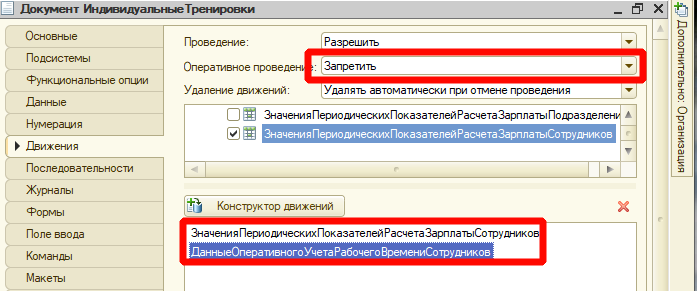
ДатаОкончанияПериода = КонецМесяца(ДобавитьМесяц(Дата, 1));

КонецПроцедуры

1. **Документ ИндивидуальныеТренировки:**



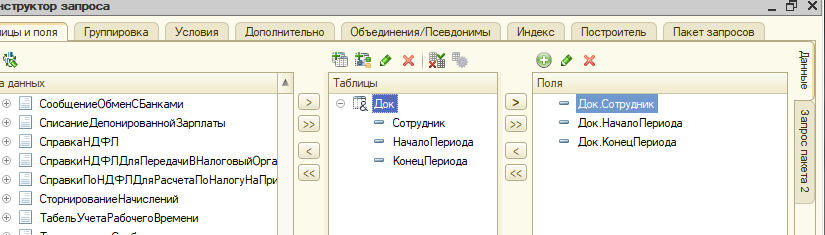
Реквизиты(1) – **Организация** - СправочникСсылка.Организации. ТаблЧасть – **Данные**(3) – **Сотрудник** (СправочникСсылка.Сотрудники), **НачалоПериода** и **КонецПериода** (Дата – состав Дата и Время)

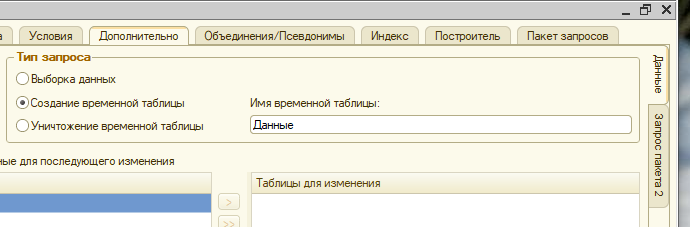


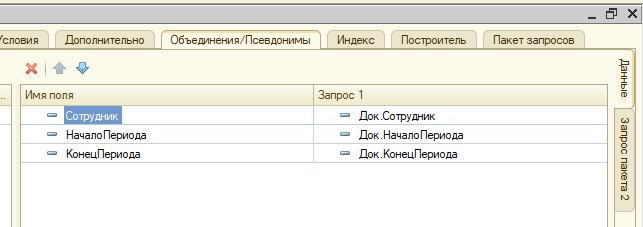
1. **Модуль документа ИндивидуальныеТренировки (движения сначала конструктором):**

**ЗАПРОС:**

**ПЕРВЫЙ ПАКЕТ:**

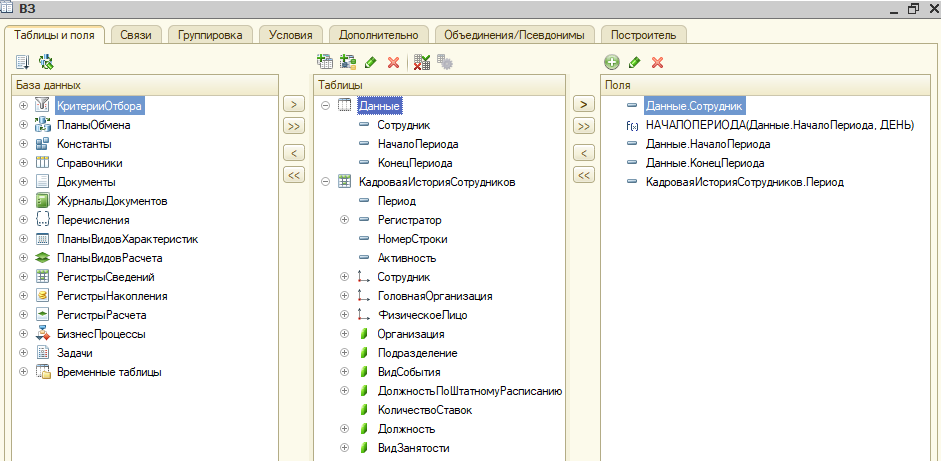


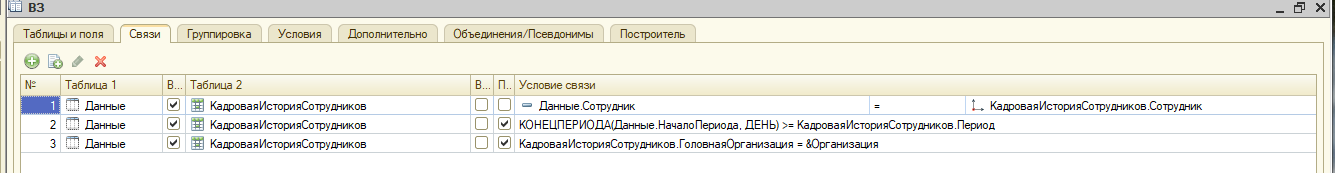


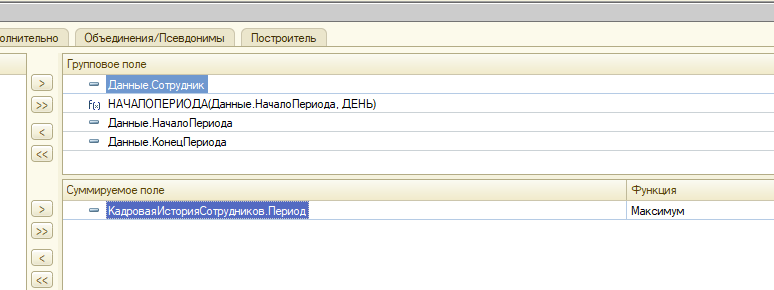


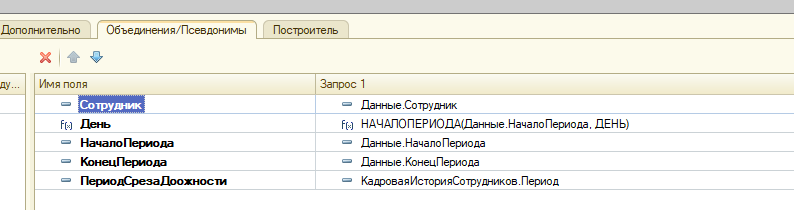
**ВТОРОЙ ПАКЕТ:**

**НАЧАЛО ВЛОЖЕННОГО ЗАПРОСА ВЗ:**

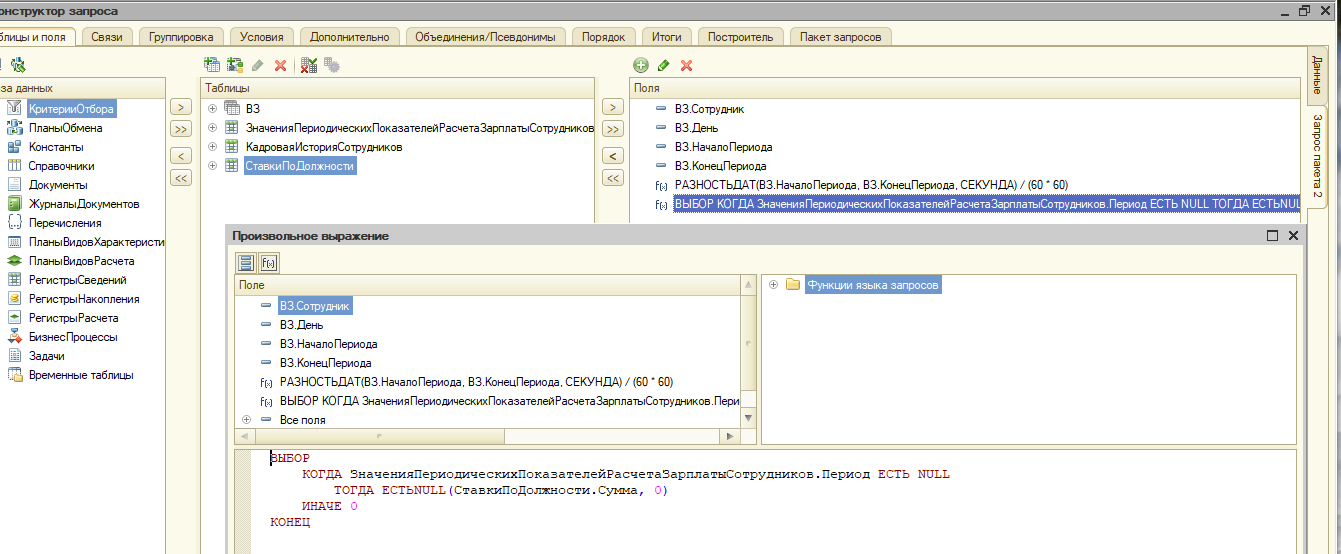


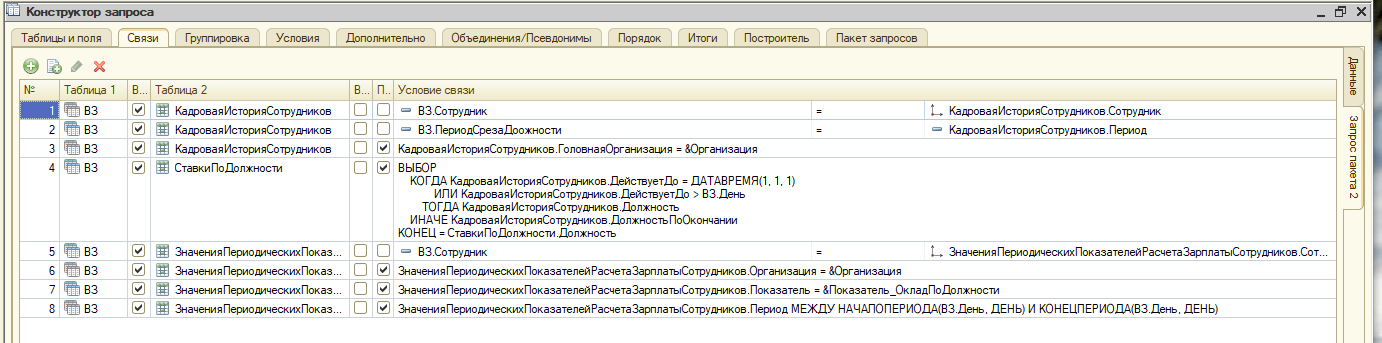


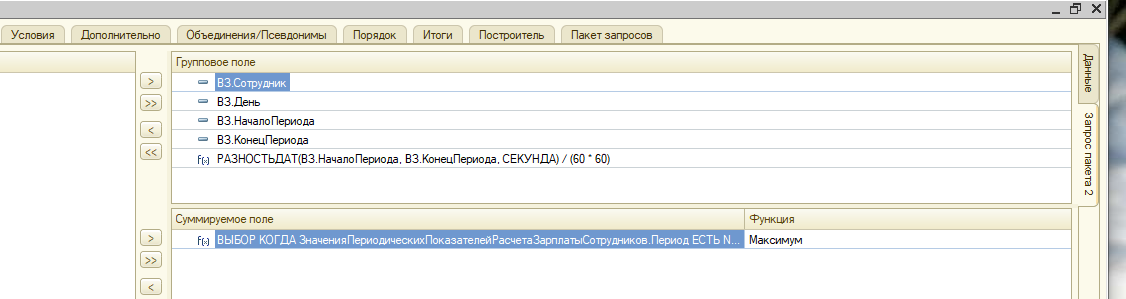


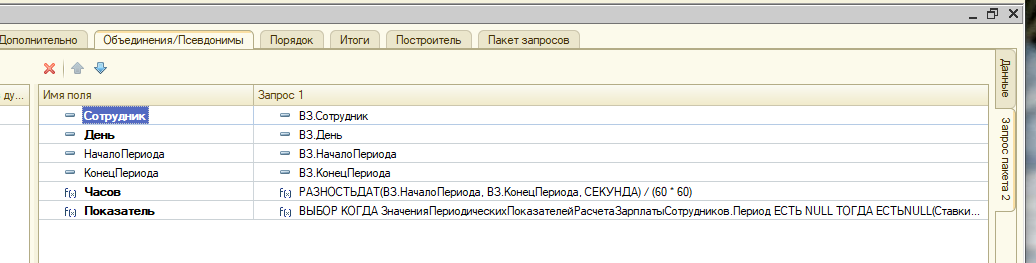


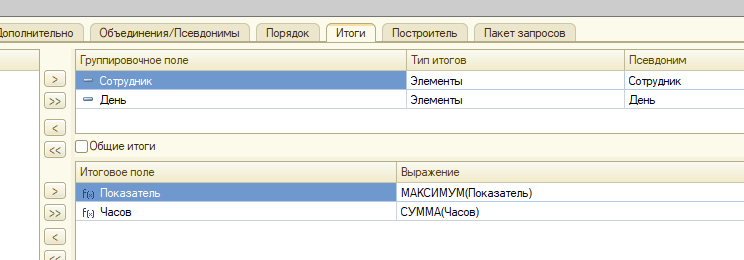
**ОКОНЧАНИЕ ВЛОЖЕННОГО ЗАПРОСА ВЗ**

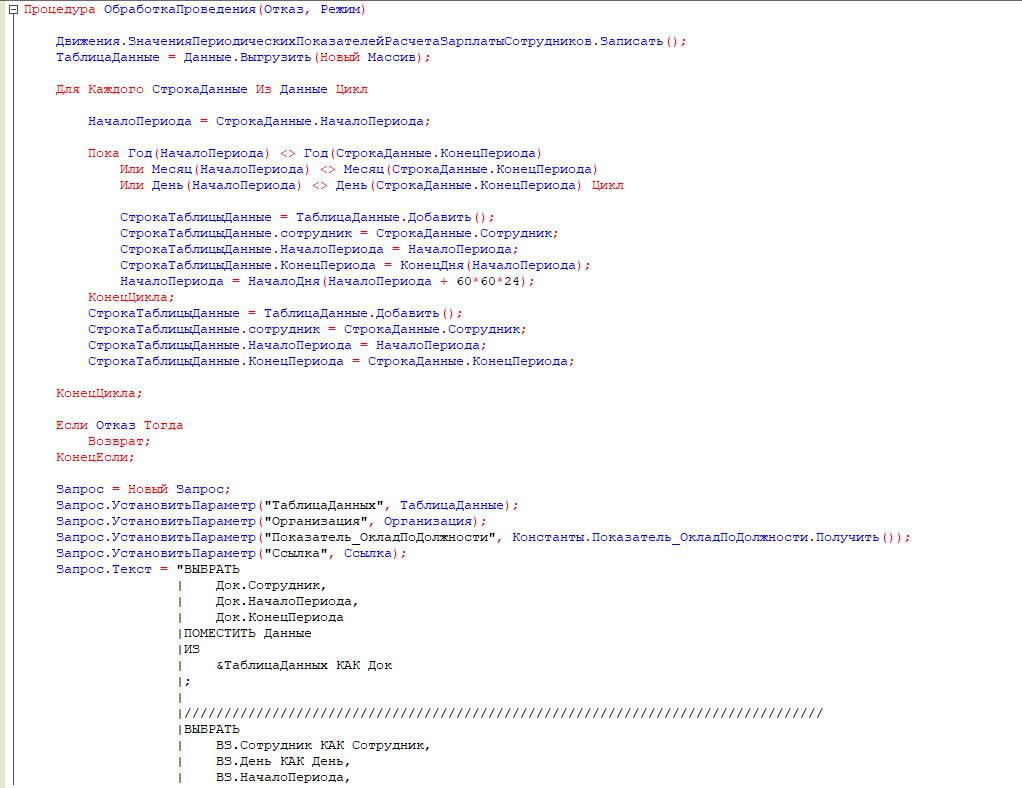


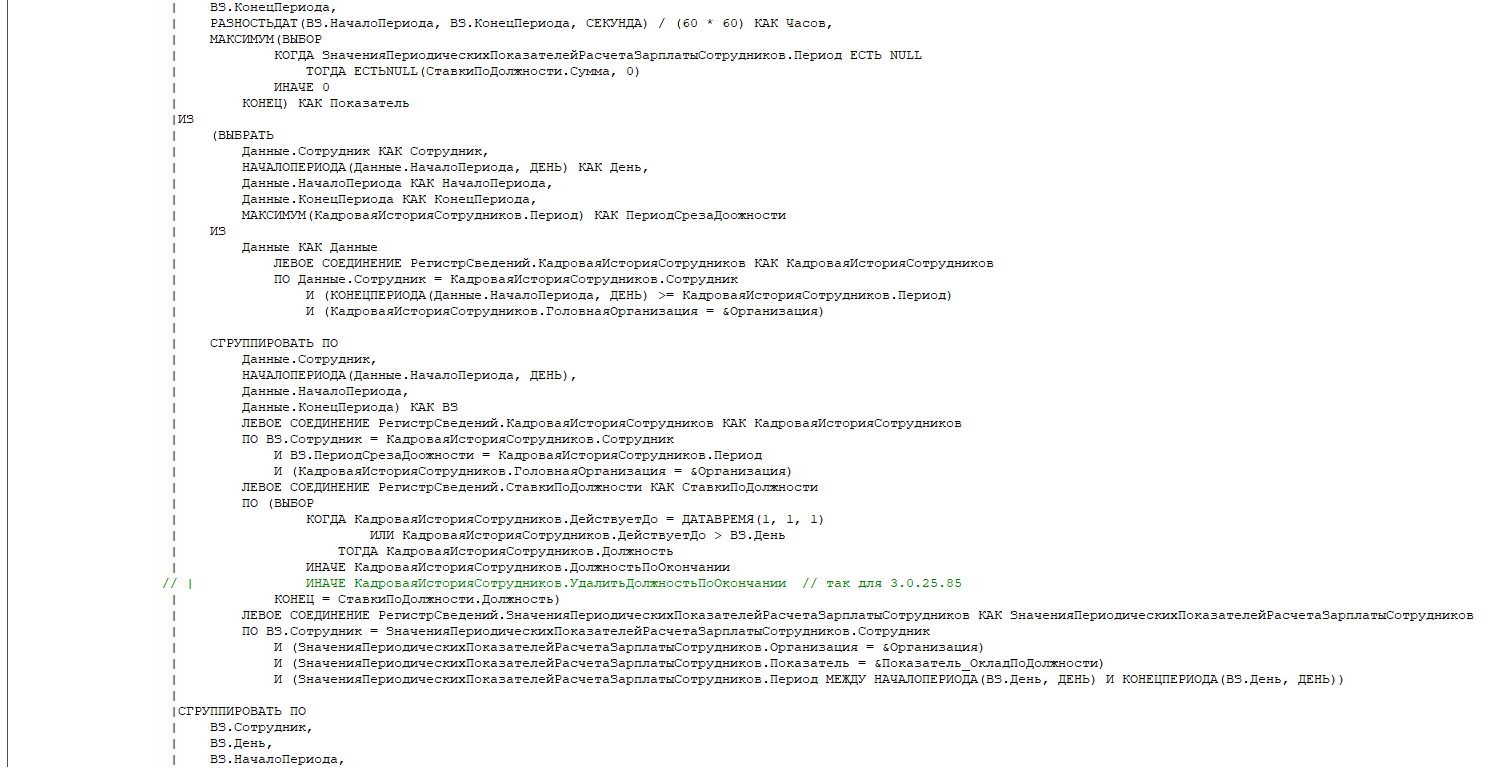




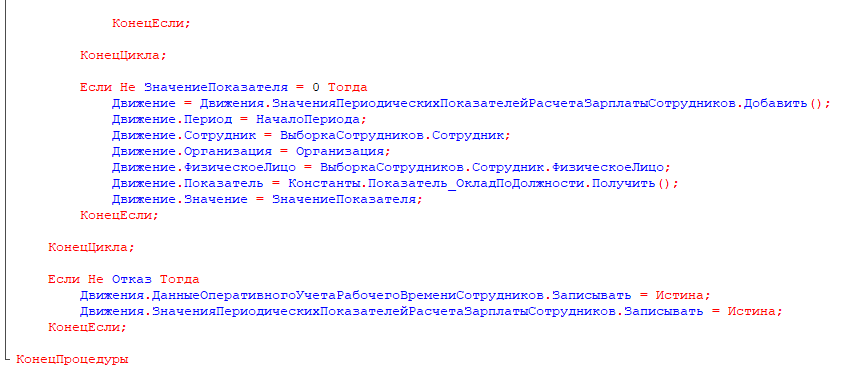












Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Записать();

ТаблицаДанные = Данные.Выгрузить(Новый Массив);

Для Каждого СтрокаДанные Из Данные Цикл

НачалоПериода = СтрокаДанные.НачалоПериода;

Пока Год(НачалоПериода) <> Год(СтрокаДанные.КонецПериода)

Или Месяц(НачалоПериода) <> Месяц(СтрокаДанные.КонецПериода)

Или День(НачалоПериода) <> День(СтрокаДанные.КонецПериода) Цикл

СтрокаТаблицыДанные = ТаблицаДанные.Добавить();

СтрокаТаблицыДанные.сотрудник = СтрокаДанные.Сотрудник;

СтрокаТаблицыДанные.НачалоПериода = НачалоПериода;

СтрокаТаблицыДанные.КонецПериода = КонецДня(НачалоПериода);

НачалоПериода = НачалоДня(НачалоПериода + 60\*60\*24);

КонецЦикла;

СтрокаТаблицыДанные = ТаблицаДанные.Добавить();

СтрокаТаблицыДанные.сотрудник = СтрокаДанные.Сотрудник;

СтрокаТаблицыДанные.НачалоПериода = НачалоПериода;

СтрокаТаблицыДанные.КонецПериода = СтрокаДанные.КонецПериода;

КонецЦикла;

Если Отказ Тогда

Возврат;

КонецЕсли;

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("ТаблицаДанных", ТаблицаДанные);

Запрос.УстановитьПараметр("Организация", Организация);

Запрос.УстановитьПараметр("Показатель\_ОкладПоДолжности", Константы.Показатель\_ОкладПоДолжности.Получить());

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.Текст = "ВЫБРАТЬ

| Док.Сотрудник,

| Док.НачалоПериода,

| Док.КонецПериода

|ПОМЕСТИТЬ Данные

|ИЗ

| &ТаблицаДанных КАК Док

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ВЗ.Сотрудник КАК Сотрудник,

| ВЗ.День КАК День,

| ВЗ.НачалоПериода,

| ВЗ.КонецПериода,

| РАЗНОСТЬДАТ(ВЗ.НачалоПериода, ВЗ.КонецПериода, СЕКУНДА) / (60 \* 60) КАК Часов,

| МАКСИМУМ(ВЫБОР

| КОГДА ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Период ЕСТЬ NULL

| ТОГДА ЕСТЬNULL(СтавкиПоДолжности.Сумма, 0)

| ИНАЧЕ 0

| КОНЕЦ) КАК Показатель

|ИЗ

| (ВЫБРАТЬ

| Данные.Сотрудник КАК Сотрудник,

| НАЧАЛОПЕРИОДА(Данные.НачалоПериода, ДЕНЬ) КАК День,

| Данные.НачалоПериода КАК НачалоПериода,

| Данные.КонецПериода КАК КонецПериода,

| МАКСИМУМ(КадроваяИсторияСотрудников.Период) КАК ПериодСрезаДоожности

| ИЗ

| Данные КАК Данные

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КадроваяИсторияСотрудников КАК КадроваяИсторияСотрудников

| ПО Данные.Сотрудник = КадроваяИсторияСотрудников.Сотрудник

| И (КОНЕЦПЕРИОДА(Данные.НачалоПериода, ДЕНЬ) >= КадроваяИсторияСотрудников.Период)

| И (КадроваяИсторияСотрудников.ГоловнаяОрганизация = &Организация)

|

| СГРУППИРОВАТЬ ПО

| Данные.Сотрудник,

| НАЧАЛОПЕРИОДА(Данные.НачалоПериода, ДЕНЬ),

| Данные.НачалоПериода,

| Данные.КонецПериода) КАК ВЗ

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КадроваяИсторияСотрудников КАК КадроваяИсторияСотрудников

| ПО ВЗ.Сотрудник = КадроваяИсторияСотрудников.Сотрудник

| И ВЗ.ПериодСрезаДоожности = КадроваяИсторияСотрудников.Период

| И (КадроваяИсторияСотрудников.ГоловнаяОрганизация = &Организация)

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.СтавкиПоДолжности КАК СтавкиПоДолжности

| ПО (ВЫБОР

| КОГДА КадроваяИсторияСотрудников.ДействуетДо = ДАТАВРЕМЯ(1, 1, 1)

| ИЛИ КадроваяИсторияСотрудников.ДействуетДо > ВЗ.День

| ТОГДА КадроваяИсторияСотрудников.Должность

| ИНАЧЕ КадроваяИсторияСотрудников.ДолжностьПоОкончании

// | ИНАЧЕ КадроваяИсторияСотрудников.УдалитьДолжностьПоОкончании // так для 3.0.25.85

| КОНЕЦ = СтавкиПоДолжности.Должность)

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников КАК ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников

| ПО ВЗ.Сотрудник = ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Сотрудник

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Организация = &Организация)

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Показатель = &Показатель\_ОкладПоДолжности)

| И (ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Период МЕЖДУ НАЧАЛОПЕРИОДА(ВЗ.День, ДЕНЬ) И КОНЕЦПЕРИОДА(ВЗ.День, ДЕНЬ))

|

|СГРУППИРОВАТЬ ПО

| ВЗ.Сотрудник,

| ВЗ.День,

| ВЗ.НачалоПериода,

| ВЗ.КонецПериода,

| РАЗНОСТЬДАТ(ВЗ.НачалоПериода, ВЗ.КонецПериода, СЕКУНДА) / (60 \* 60)

|ИТОГИ

| СУММА(Часов),

| МАКСИМУМ(Показатель)

|ПО

| Сотрудник,

| День

|";

ВыборкаСотрудников = Запрос.Выполнить().Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Пока ВыборкаСотрудников.Следующий() Цикл

ЗначениеПоказателя = 0;

НачалоПериода = Неопределено;

ВыборкаДень = ВыборкаСотрудников.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

Пока ВыборкаДень.Следующий() Цикл

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = ВыборкаДень.День;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(ВыборкаДень.День);

Движение.ВидУчетаВремени = Константы.ВидВремени\_ИндивидуальныеТренировки.Получить();

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДополнительноеВнутрисменноеВремя;

Движение.Часы = ВыборкаДень.Часов;

Если ЗначениеПоказателя = 0 Тогда

ЗначениеПоказателя = ВыборкаДень.Показатель;

НачалоПериода = ВыборкаДень.День;

Продолжить;

КонецЕсли;

Если ЗначениеПоказателя <> ВыборкаДень.Показатель И Не ЗначениеПоказателя = 0 Тогда

Движение = Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Добавить();

Движение.Период = НачалоПериода;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.Организация = Организация;

Движение.ФизическоеЛицо = ВыборкаСотрудников.Сотрудник.ФизическоеЛицо;

Движение.Показатель = Константы.Показатель\_ОкладПоДолжности.Получить();

Движение.Значение = ЗначениеПоказателя;

Движение.ДействуетДо = ВыборкаДень.День;

ЗначениеПоказателя = ВыборкаДень.Показатель;

НачалоПериода = ВыборкаДень.День;

Продолжить;

КонецЕсли;

КонецЦикла;

Если Не ЗначениеПоказателя = 0 Тогда

Движение = Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Добавить();

Движение.Период = НачалоПериода;

Движение.Сотрудник = ВыборкаСотрудников.Сотрудник;

Движение.Организация = Организация;

Движение.ФизическоеЛицо = ВыборкаСотрудников.Сотрудник.ФизическоеЛицо;

Движение.Показатель = Константы.Показатель\_ОкладПоДолжности.Получить();

Движение.Значение = ЗначениеПоказателя;

КонецЕсли;

КонецЦикла;

Если Не Отказ Тогда

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников.Записывать = Истина;

КонецЕсли;

КонецПроцедуры

1. **Отчет ОплатаТренеров:**

**ЗАПРОС:**

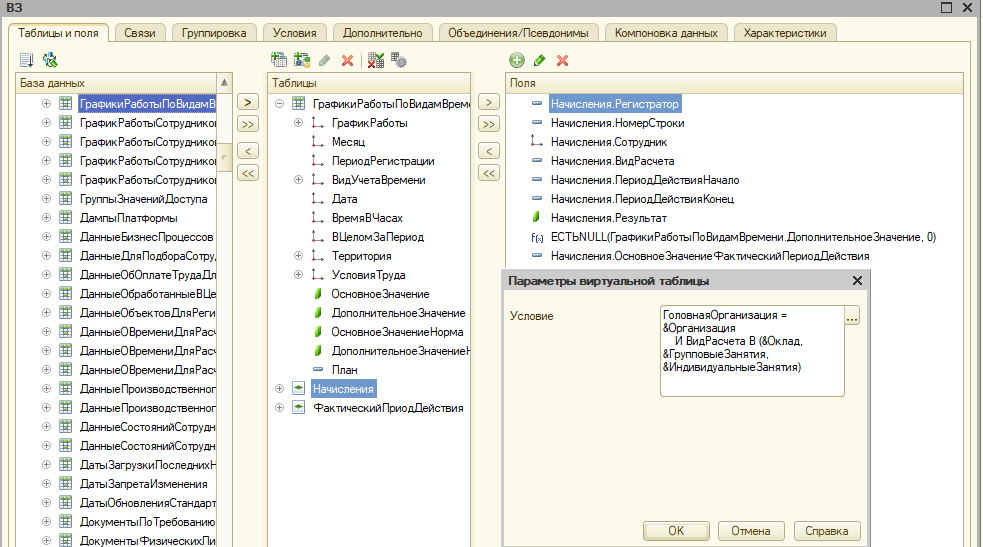
**ПЕРВЫЙ ПАКЕТ:**

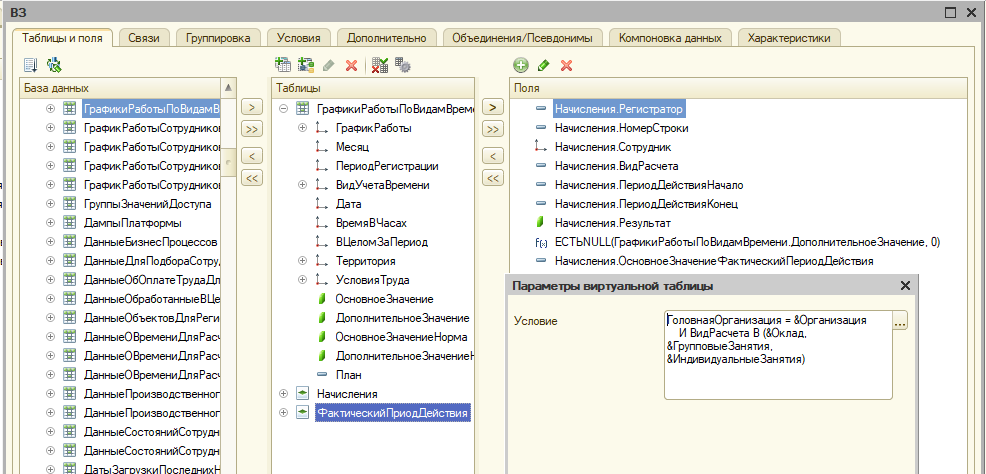
**ЗАПРОС 1 из первого пакета:**

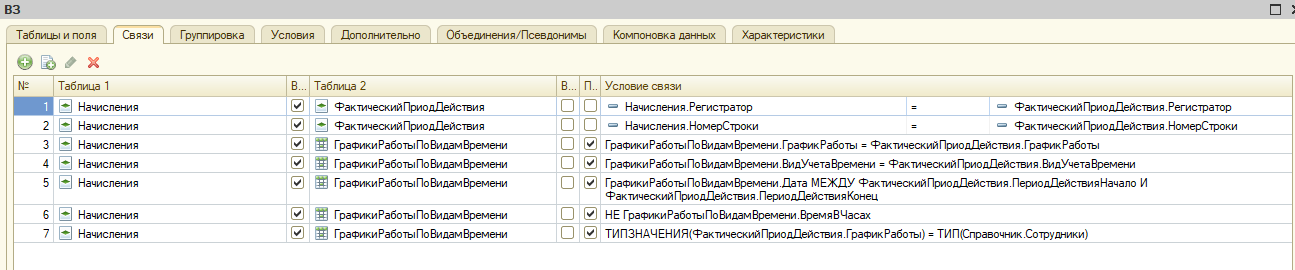
**НАЧАЛО ВЛОЖЕННОГО ЗАПРОСА ВЗ:**

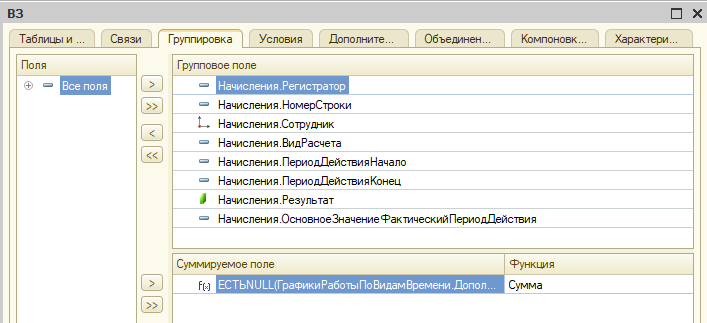
**ПЕРВЫЙ ПАКЕТ:**

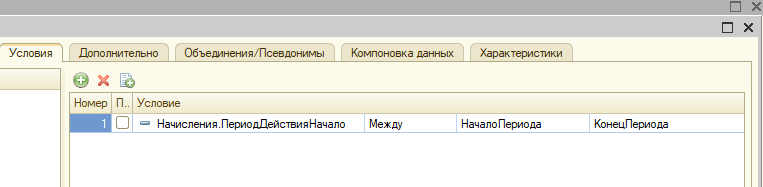
**НачисленияФактическийПериодДействия** – переименовать в **ФактическийПриодДействия НачисленияДанныеГрафика –** переименовать в **Начисления**

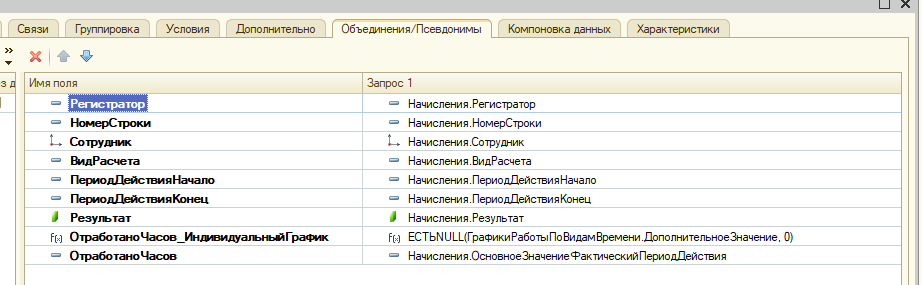




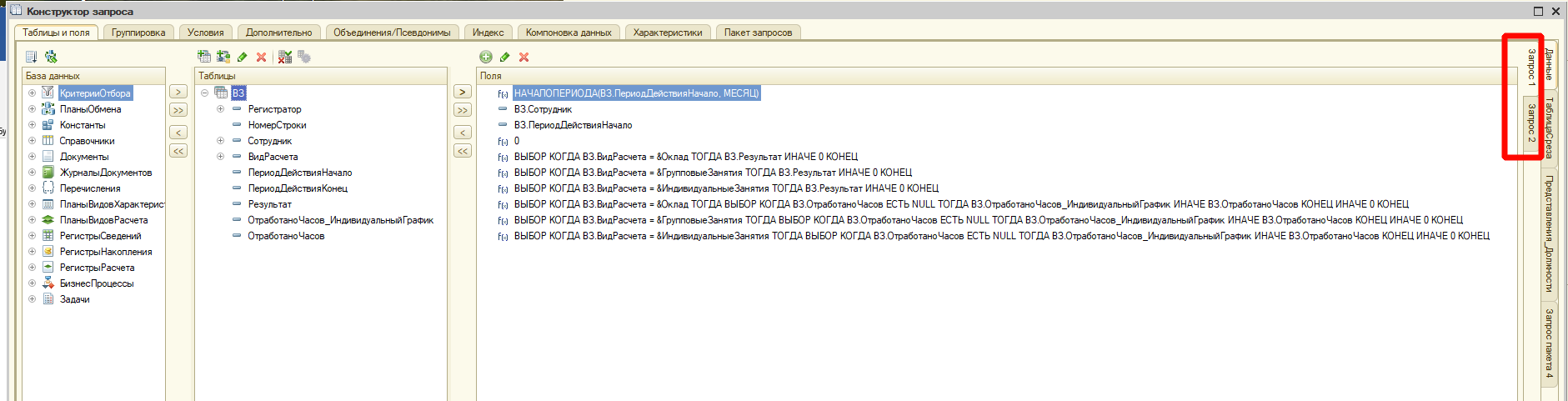








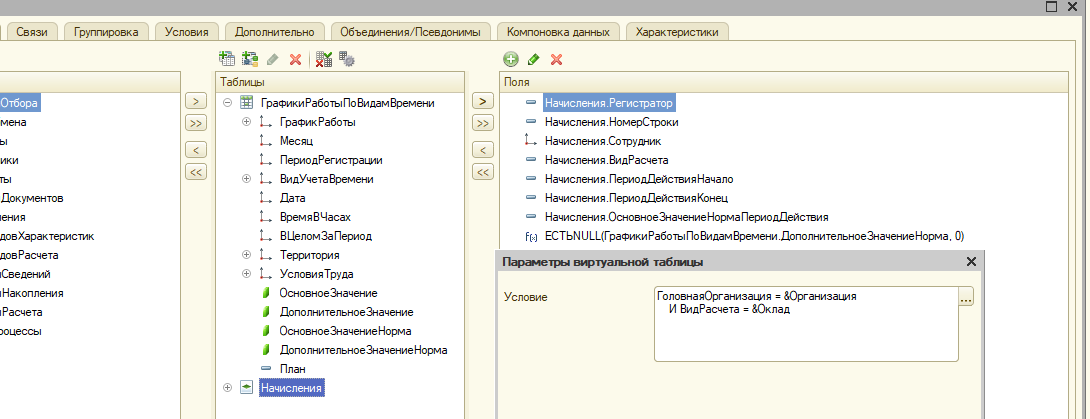
**ОКОНЧАНИЕ ВЛОЖЕННОГО ЗАПРОСА ВЗ:**

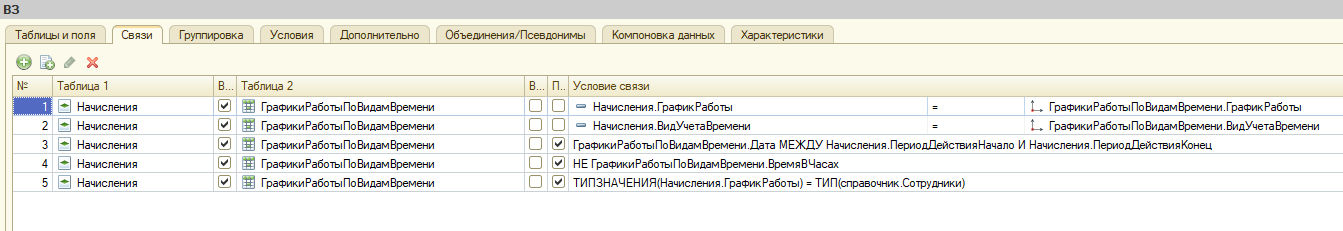


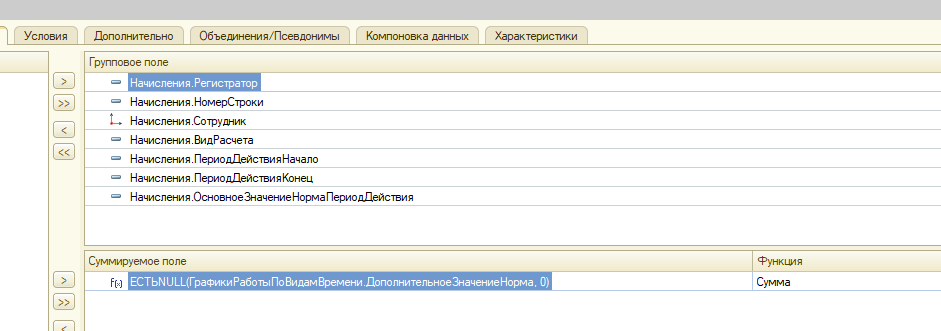
**ЗАПРОС 2 из первого пакета:**

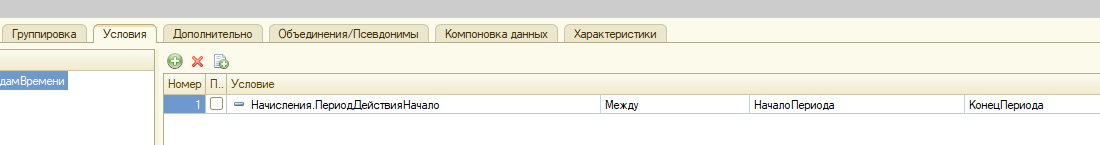
**НАЧАЛО ВЛОЖЕННОГО ЗАПРОСА ВЗ:**

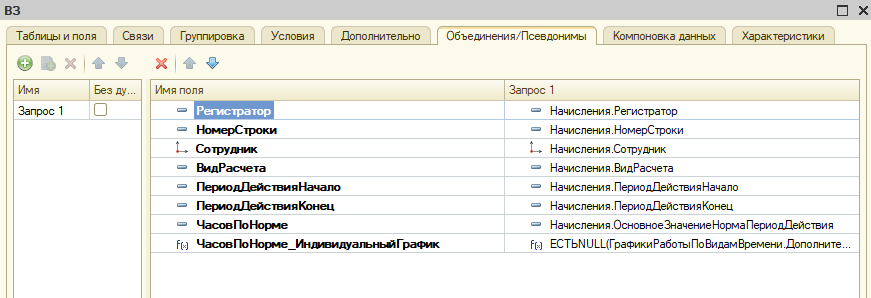
Здесь **Начисления** – тоже переименовали – было **НачисленияДанныеГрафика**.



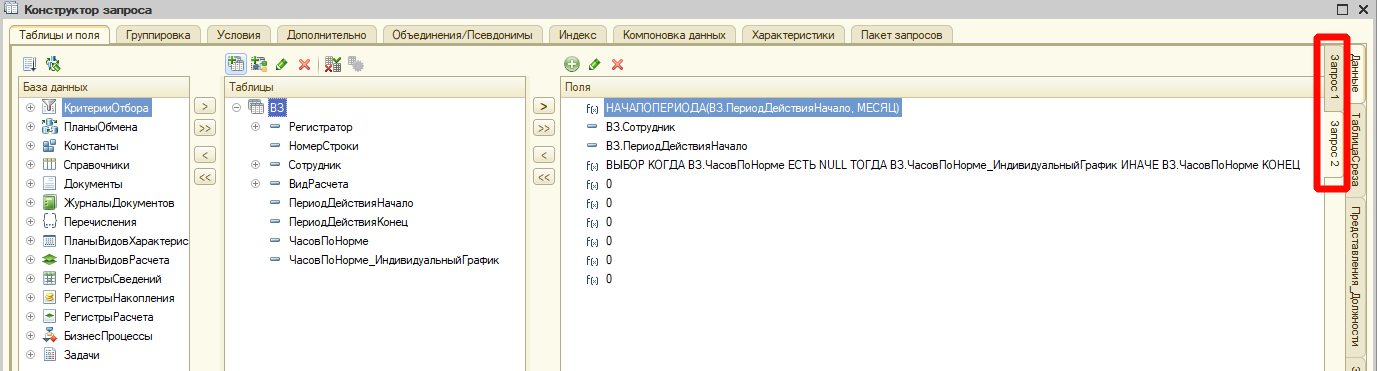




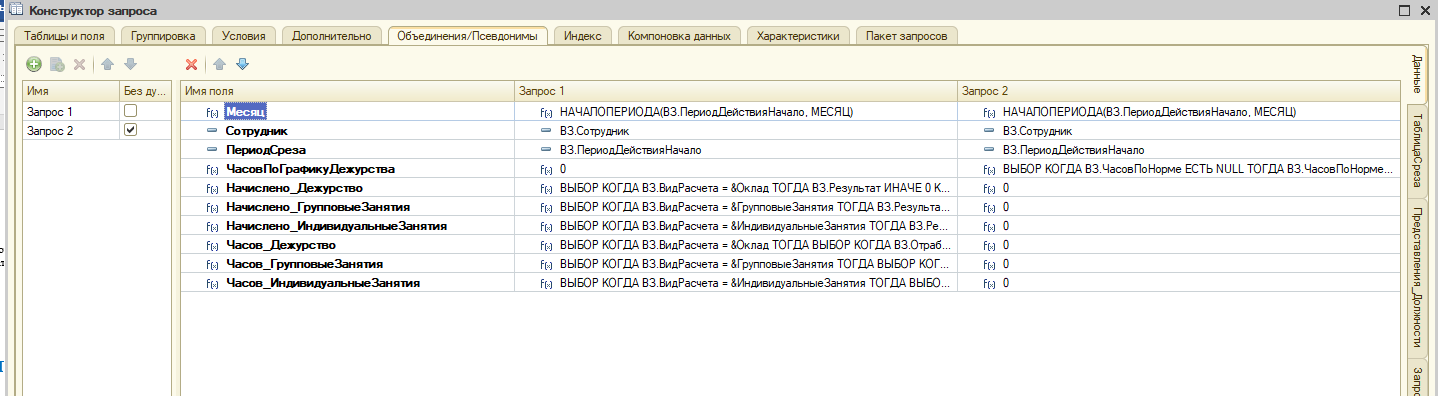


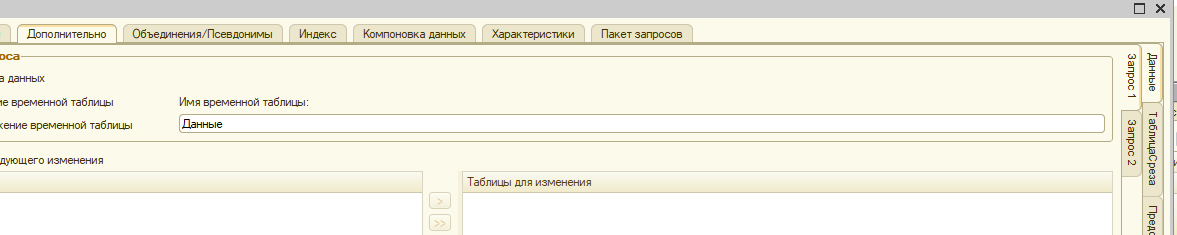


**ОКОНЧАНИЕ ВЛОЖЕННОГО ЗАПРОСА ВЗ:**

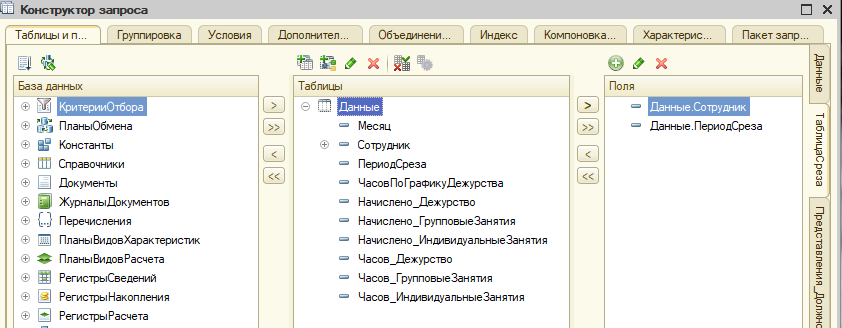


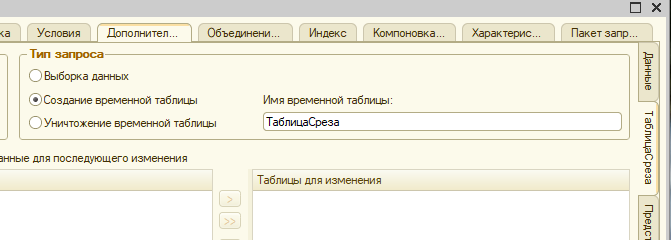
**ОБЪЕДИНЕНИЕ (ЭТО ВСЕ ПЕРВЫЙ ПАКЕТ):**

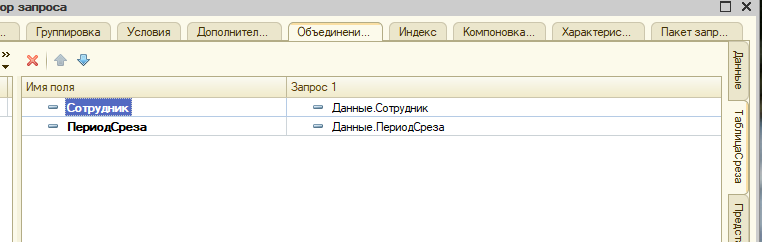




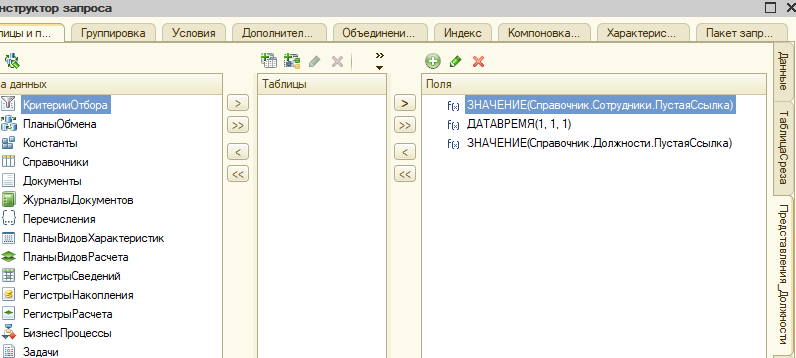
**ВТОРОЙ ПАКЕТ:**

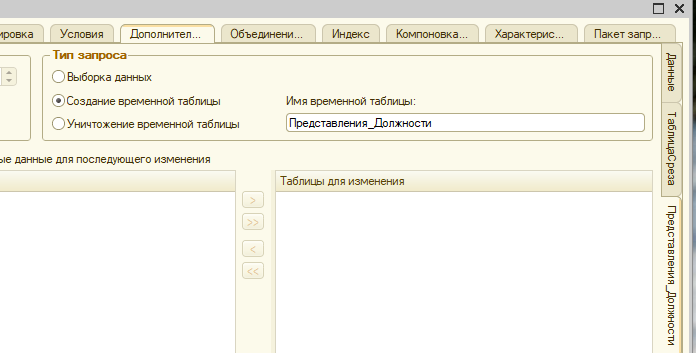


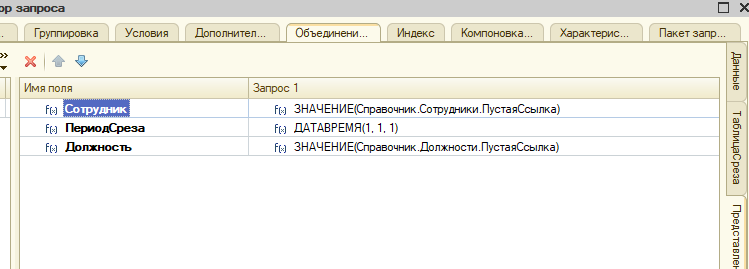




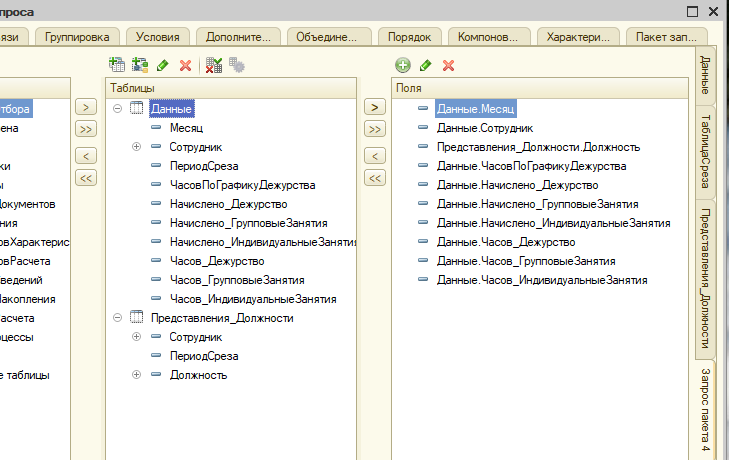
**ТРЕТИЙ ПАКЕТ:**

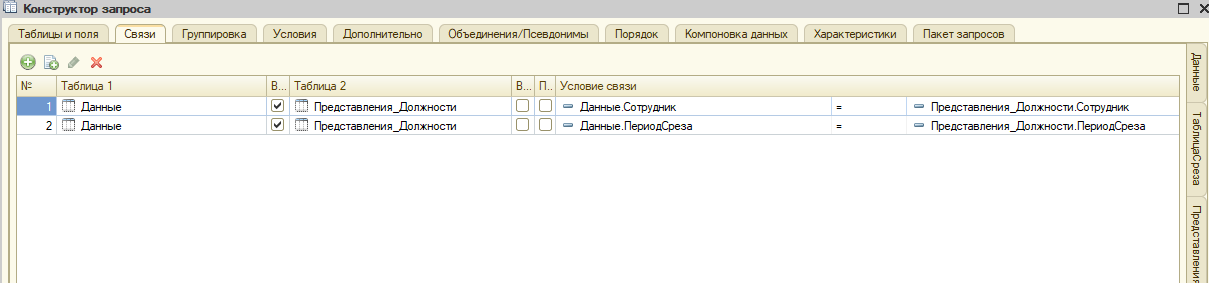


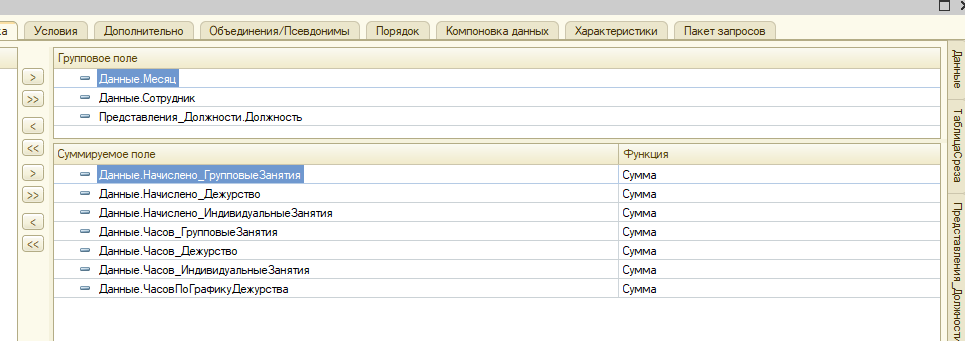


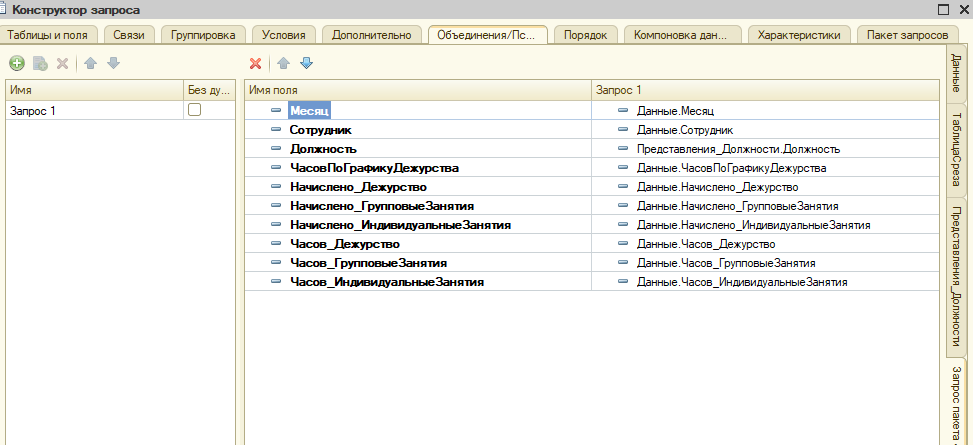


**ЧЕТВЕРТЫЙ ПАКЕТ:**









ВЫБРАТЬ

НАЧАЛОПЕРИОДА(ВЗ.ПериодДействияНачало, МЕСЯЦ) КАК Месяц,

ВЗ.Сотрудник КАК Сотрудник,

ВЗ.ПериодДействияНачало КАК ПериодСреза,

0 КАК ЧасовПоГрафикуДежурства,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &Оклад

ТОГДА ВЗ.Результат

ИНАЧЕ 0

КОНЕЦ КАК Начислено\_Дежурство,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &ГрупповыеЗанятия

ТОГДА ВЗ.Результат

ИНАЧЕ 0

КОНЕЦ КАК Начислено\_ГрупповыеЗанятия,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &ИндивидуальныеЗанятия

ТОГДА ВЗ.Результат

ИНАЧЕ 0

КОНЕЦ КАК Начислено\_ИндивидуальныеЗанятия,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &Оклад

ТОГДА ВЫБОР

КОГДА ВЗ.ОтработаноЧасов ЕСТЬ NULL

ТОГДА ВЗ.ОтработаноЧасов\_ИндивидуальныйГрафик

ИНАЧЕ ВЗ.ОтработаноЧасов

КОНЕЦ

ИНАЧЕ 0

КОНЕЦ КАК Часов\_Дежурство,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &ГрупповыеЗанятия

ТОГДА ВЫБОР

КОГДА ВЗ.ОтработаноЧасов ЕСТЬ NULL

ТОГДА ВЗ.ОтработаноЧасов\_ИндивидуальныйГрафик

ИНАЧЕ ВЗ.ОтработаноЧасов

КОНЕЦ

ИНАЧЕ 0

КОНЕЦ КАК Часов\_ГрупповыеЗанятия,

ВЫБОР

КОГДА ВЗ.ВидРасчета = &ИндивидуальныеЗанятия

ТОГДА ВЫБОР

КОГДА ВЗ.ОтработаноЧасов ЕСТЬ NULL

ТОГДА ВЗ.ОтработаноЧасов\_ИндивидуальныйГрафик

ИНАЧЕ ВЗ.ОтработаноЧасов

КОНЕЦ

ИНАЧЕ 0

КОНЕЦ КАК Часов\_ИндивидуальныеЗанятия

ПОМЕСТИТЬ Данные

ИЗ

(ВЫБРАТЬ

Начисления.Регистратор КАК Регистратор,

Начисления.НомерСтроки КАК НомерСтроки,

Начисления.Сотрудник КАК Сотрудник,

Начисления.ВидРасчета КАК ВидРасчета,

Начисления.ПериодДействияНачало КАК ПериодДействияНачало,

Начисления.ПериодДействияКонец КАК ПериодДействияКонец,

Начисления.Результат КАК Результат,

СУММА(ЕСТЬNULL(ГрафикиРаботыПоВидамВремени.ДополнительноеЗначение, 0)) КАК ОтработаноЧасов\_ИндивидуальныйГрафик,

Начисления.ОсновноеЗначениеФактическийПериодДействия КАК ОтработаноЧасов

ИЗ

РегистрРасчета.Начисления.ДанныеГрафика(

ГоловнаяОрганизация = &Организация

И ВидРасчета В (&Оклад, &ГрупповыеЗанятия, &ИндивидуальныеЗанятия)) КАК Начисления

ЛЕВОЕ СОЕДИНЕНИЕ РегистрРасчета.Начисления.ФактическийПериодДействия(

ГоловнаяОрганизация = &Организация

И ВидРасчета В (&Оклад, &ГрупповыеЗанятия, &ИндивидуальныеЗанятия)) КАК ФактическийПриодДействия

ПО Начисления.Регистратор = ФактическийПриодДействия.Регистратор

И Начисления.НомерСтроки = ФактическийПриодДействия.НомерСтроки

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ГрафикиРаботыПоВидамВремени КАК ГрафикиРаботыПоВидамВремени

ПО (ГрафикиРаботыПоВидамВремени.ГрафикРаботы = ФактическийПриодДействия.ГрафикРаботы)

И (ГрафикиРаботыПоВидамВремени.ВидУчетаВремени = ФактическийПриодДействия.ВидУчетаВремени)

И (ГрафикиРаботыПоВидамВремени.Дата МЕЖДУ ФактическийПриодДействия.ПериодДействияНачало И ФактическийПриодДействия.ПериодДействияКонец)

И (НЕ ГрафикиРаботыПоВидамВремени.ВремяВЧасах)

И (ТИПЗНАЧЕНИЯ(ФактическийПриодДействия.ГрафикРаботы) = ТИП(Справочник.Сотрудники))

ГДЕ

Начисления.ПериодДействияНачало МЕЖДУ &НачалоПериода И &КонецПериода

СГРУППИРОВАТЬ ПО

Начисления.Регистратор,

Начисления.НомерСтроки,

Начисления.Сотрудник,

Начисления.ВидРасчета,

Начисления.ПериодДействияНачало,

Начисления.ПериодДействияКонец,

Начисления.Результат,

Начисления.ОсновноеЗначениеФактическийПериодДействия) КАК ВЗ

ОБЪЕДИНИТЬ

ВЫБРАТЬ

НАЧАЛОПЕРИОДА(ВЗ.ПериодДействияНачало, МЕСЯЦ),

ВЗ.Сотрудник,

ВЗ.ПериодДействияНачало,

ВЫБОР

КОГДА ВЗ.ЧасовПоНорме ЕСТЬ NULL

ТОГДА ВЗ.ЧасовПоНорме\_ИндивидуальныйГрафик

ИНАЧЕ ВЗ.ЧасовПоНорме

КОНЕЦ,

0,

0,

0,

0,

0,

0

ИЗ

(ВЫБРАТЬ

Начисления.Регистратор КАК Регистратор,

Начисления.НомерСтроки КАК НомерСтроки,

Начисления.Сотрудник КАК Сотрудник,

Начисления.ВидРасчета КАК ВидРасчета,

Начисления.ПериодДействияНачало КАК ПериодДействияНачало,

Начисления.ПериодДействияКонец КАК ПериодДействияКонец,

Начисления.ОсновноеЗначениеНормаПериодДействия КАК ЧасовПоНорме,

СУММА(ЕСТЬNULL(ГрафикиРаботыПоВидамВремени.ДополнительноеЗначениеНорма, 0)) КАК ЧасовПоНорме\_ИндивидуальныйГрафик

ИЗ

РегистрРасчета.Начисления.ДанныеГрафика(

ГоловнаяОрганизация = &Организация

И ВидРасчета = &Оклад) КАК Начисления

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ГрафикиРаботыПоВидамВремени КАК ГрафикиРаботыПоВидамВремени

ПО Начисления.ГрафикРаботы = ГрафикиРаботыПоВидамВремени.ГрафикРаботы

И Начисления.ВидУчетаВремени = ГрафикиРаботыПоВидамВремени.ВидУчетаВремени

И (ГрафикиРаботыПоВидамВремени.Дата МЕЖДУ Начисления.ПериодДействияНачало И Начисления.ПериодДействияКонец)

И (НЕ ГрафикиРаботыПоВидамВремени.ВремяВЧасах)

И (ТИПЗНАЧЕНИЯ(Начисления.ГрафикРаботы) = ТИП(справочник.Сотрудники))

ГДЕ

Начисления.ПериодДействияНачало МЕЖДУ &НачалоПериода И &КонецПериода

СГРУППИРОВАТЬ ПО

Начисления.Регистратор,

Начисления.НомерСтроки,

Начисления.Сотрудник,

Начисления.ВидРасчета,

Начисления.ПериодДействияНачало,

Начисления.ПериодДействияКонец,

Начисления.ОсновноеЗначениеНормаПериодДействия) КАК ВЗ

;

////////////////////////////////////////////////////////////////////////////////

ВЫБРАТЬ

Данные.Сотрудник КАК Сотрудник,

Данные.ПериодСреза КАК ПериодСреза

ПОМЕСТИТЬ ТаблицаСреза

ИЗ

Данные КАК Данные

;

////////////////////////////////////////////////////////////////////////////////

ВЫБРАТЬ

ЗНАЧЕНИЕ(Справочник.Сотрудники.ПустаяСсылка) КАК Сотрудник,

ДАТАВРЕМЯ(1, 1, 1) КАК ПериодСреза,

ЗНАЧЕНИЕ(Справочник.Должности.ПустаяСсылка) КАК Должность

ПОМЕСТИТЬ Представления\_Должности

;

////////////////////////////////////////////////////////////////////////////////

ВЫБРАТЬ

Данные.Месяц КАК Месяц,

Данные.Сотрудник КАК Сотрудник,

Представления\_Должности.Должность КАК Должность,

СУММА(Данные.ЧасовПоГрафикуДежурства) КАК ЧасовПоГрафикуДежурства,

СУММА(Данные.Начислено\_Дежурство) КАК Начислено\_Дежурство,

СУММА(Данные.Начислено\_ГрупповыеЗанятия) КАК Начислено\_ГрупповыеЗанятия,

СУММА(Данные.Начислено\_ИндивидуальныеЗанятия) КАК Начислено\_ИндивидуальныеЗанятия,

СУММА(Данные.Часов\_Дежурство) КАК Часов\_Дежурство,

СУММА(Данные.Часов\_ГрупповыеЗанятия) КАК Часов\_ГрупповыеЗанятия,

СУММА(Данные.Часов\_ИндивидуальныеЗанятия) КАК Часов\_ИндивидуальныеЗанятия

ИЗ

Данные КАК Данные

ЛЕВОЕ СОЕДИНЕНИЕ Представления\_Должности КАК Представления\_Должности

ПО Данные.Сотрудник = Представления\_Должности.Сотрудник

И Данные.ПериодСреза = Представления\_Должности.ПериодСреза

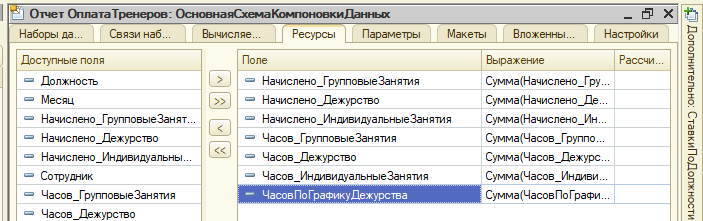
СГРУППИРОВАТЬ ПО

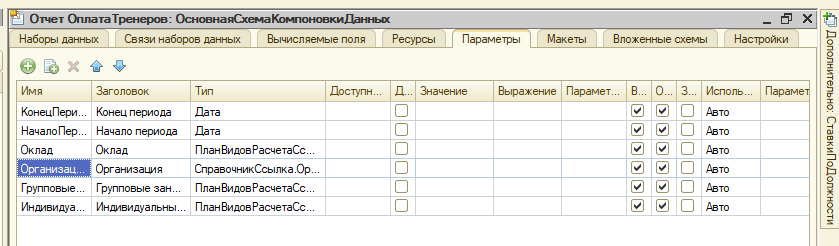
Данные.Месяц,

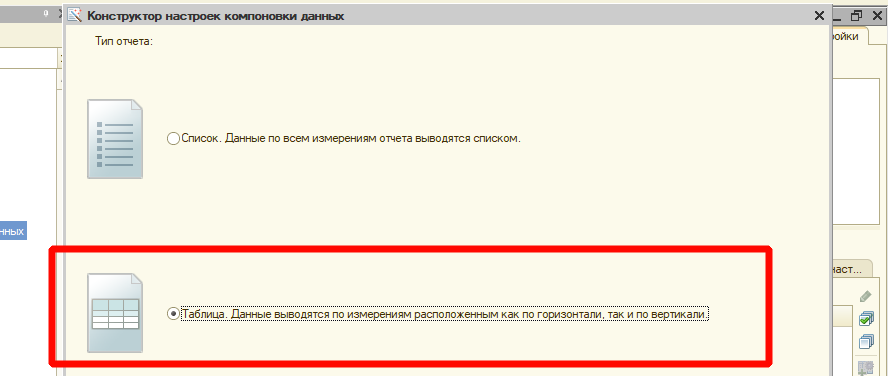
Данные.Сотрудник,

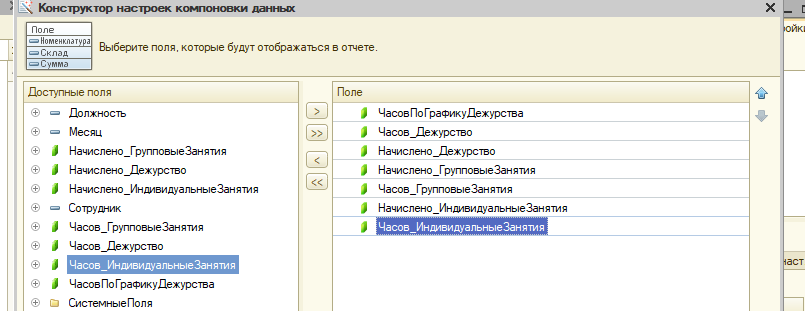
Представления\_Должности.Должность

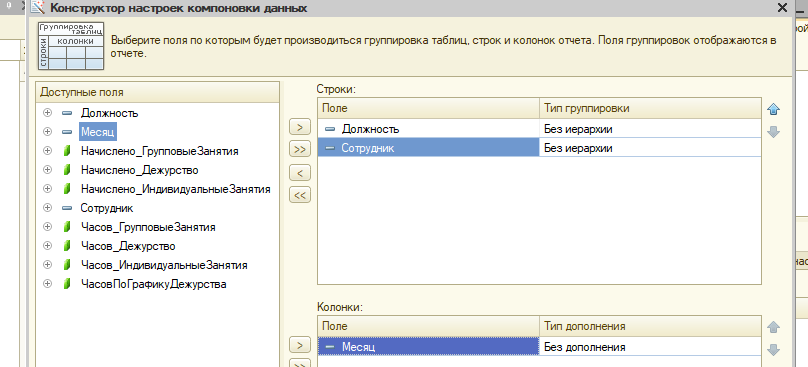
**Далее отчет в СКД:**



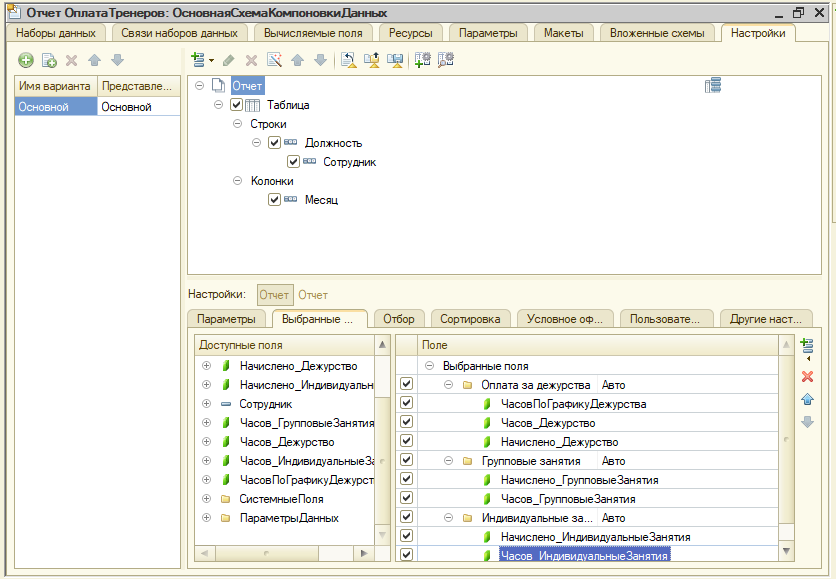




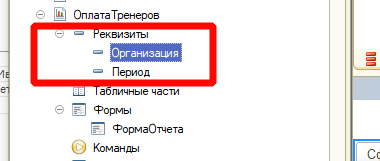




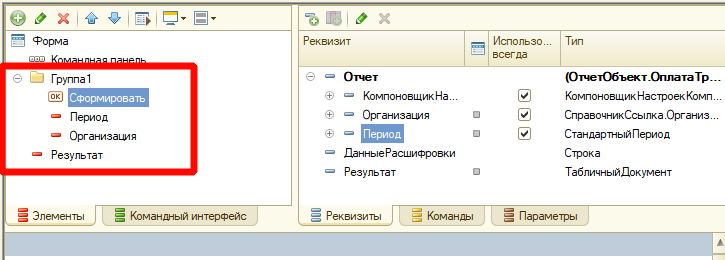
**Создаем группы:**



**Добавляем реквизиты:**



**Создаем форму отчета, кнопка «Сформировать» клиент-сервер:**



**Модуль формы отчета:**



&НаКлиенте

Процедура Сформировать(Команда)

СформироватьНаСервере();

КонецПроцедуры

&НаСервере

Процедура СформироватьНаСервере()

Настройки = Отчет.КомпоновщикНастроек.ПолучитьНастройки();

Для Каждого ПараметрДанных Из Настройки.ПараметрыДанных.Элементы Цикл

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("КонецПериода") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = КонецМесяца(Отчет.Период.ДатаОкончания);

КонецЕсли;

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("НачалоПериода") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = НачалоМесяца(Отчет.Период.ДатаНачала);

КонецЕсли;

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("Оклад") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = Константы.Начисление\_Оклад.Получить();

КонецЕсли;

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("Организация") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = Отчет.Организация;

КонецЕсли;

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("ГрупповыеЗанятия") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = Константы.Начисление\_ГрупповыеЗанятия.Получить();

КонецЕсли;

Если ПараметрДанных.Параметр = Новый ПараметрКомпоновкиДанных("ИндивидуальныеЗанятия") Тогда

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = Константы.Начисление\_ИндивидуальныеЗанятия.Получить();

КонецЕсли;

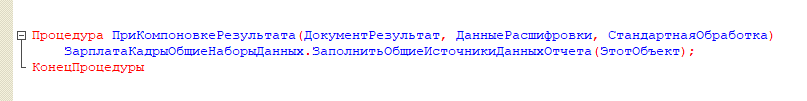
КонецЦикла;

Отчет.КомпоновщикНастроек.ЗагрузитьНастройки(Настройки);

СкомпоноватьРезультат();

КонецПроцедуры

**Модуль отчета:**



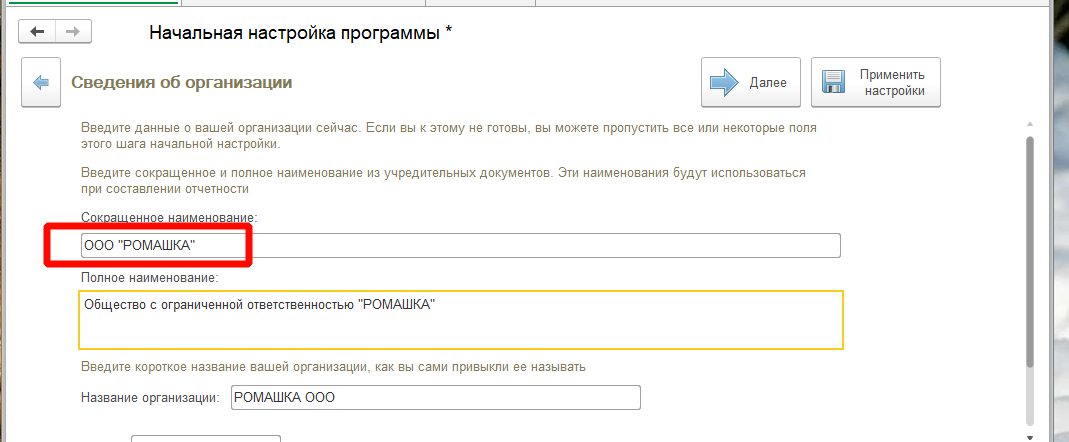
Процедура ПриКомпоновкеРезультата(ДокументРезультат, ДанныеРасшифровки, СтандартнаяОбработка)

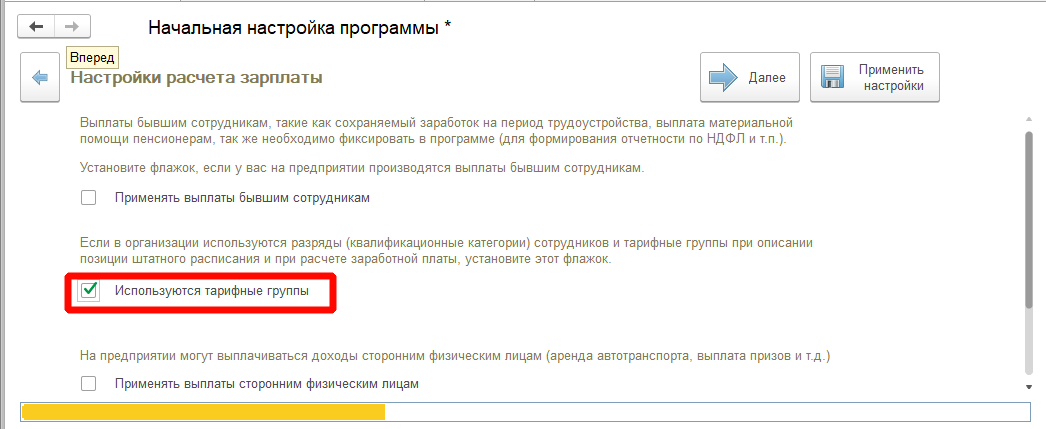
ЗарплатаКадрыОбщиеНаборыДанных.ЗаполнитьОбщиеИсточникиДанныхОтчета(ЭтотОбъект);

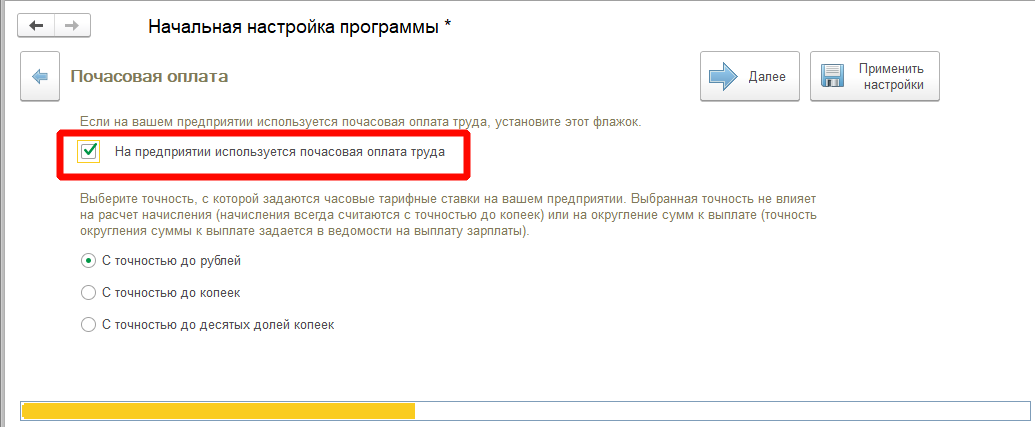
КонецПроцедуры

**III Ввод контрольного примера и проверка решения:**

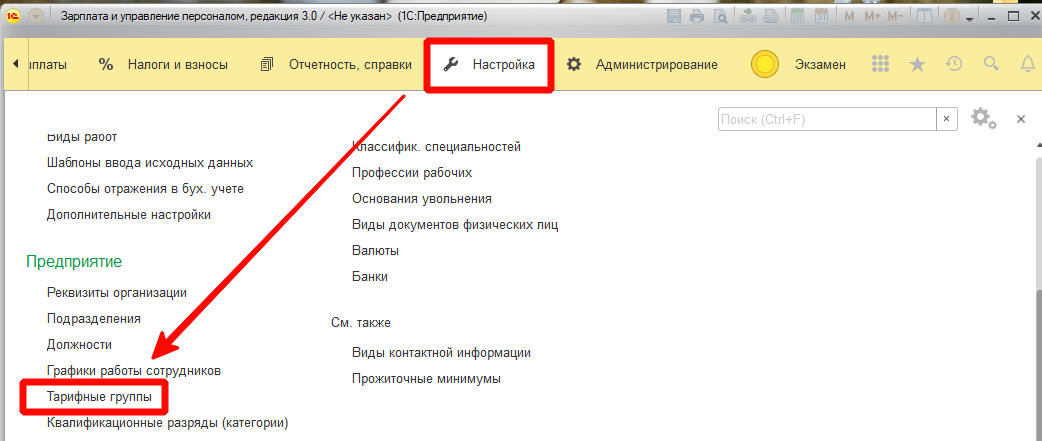
1. **Делаем начальную настройку программы:**

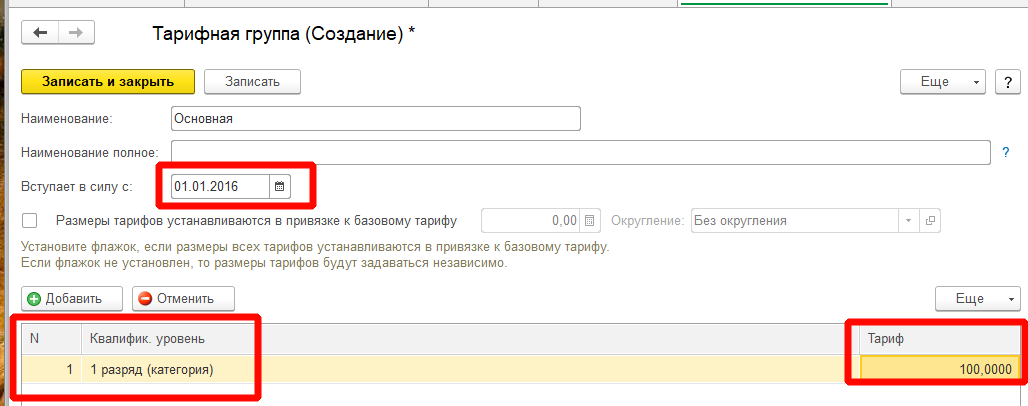




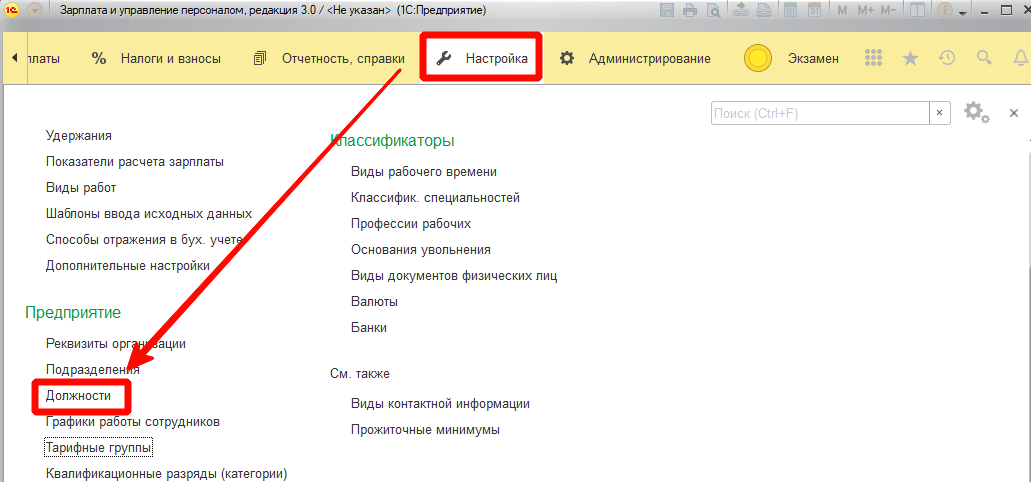


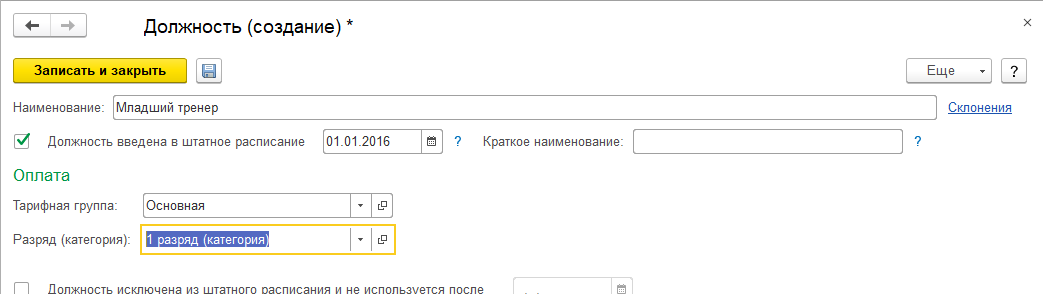
1. **Создаем тарифную группу:**

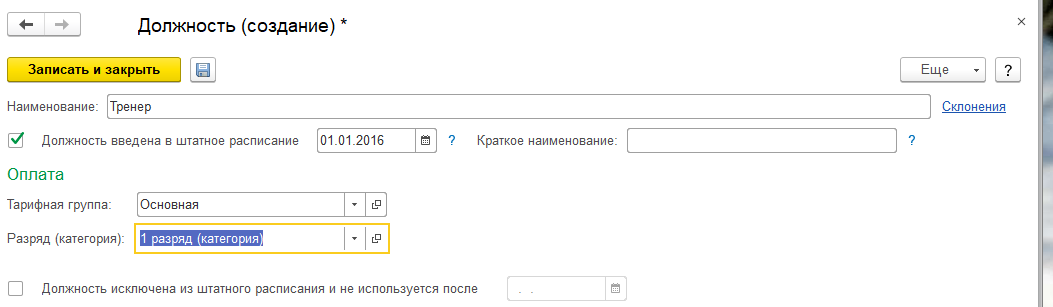


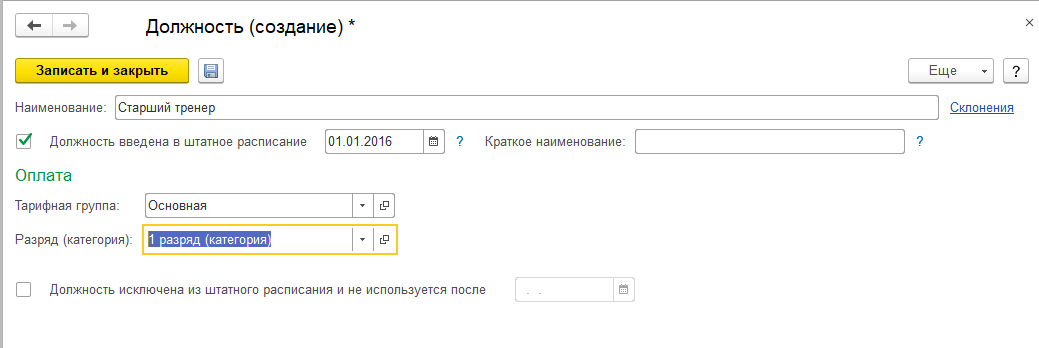


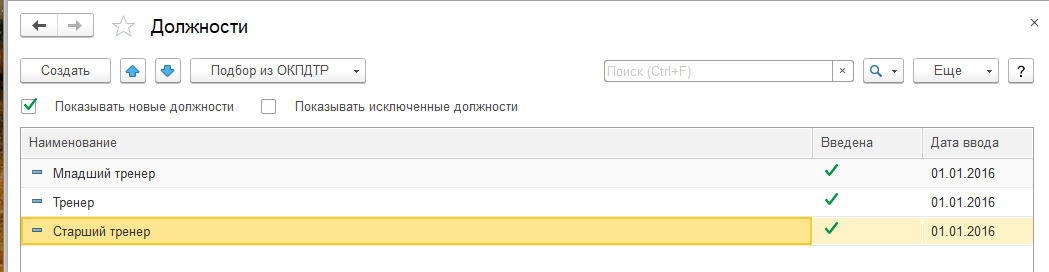
1. **Создаем 3 должности:**



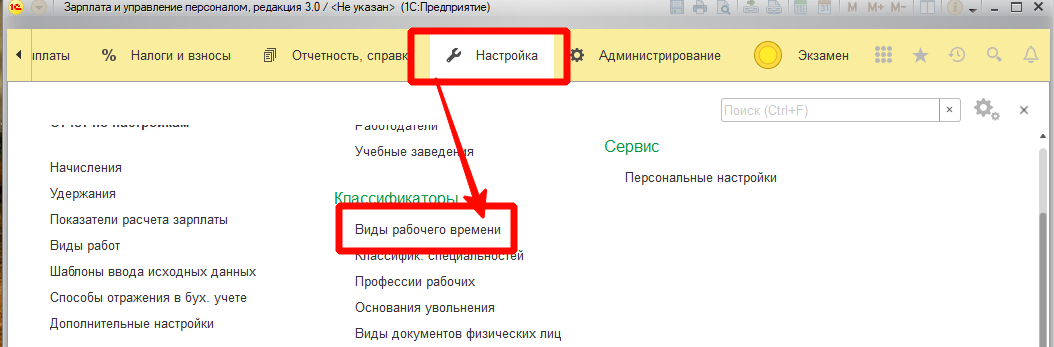


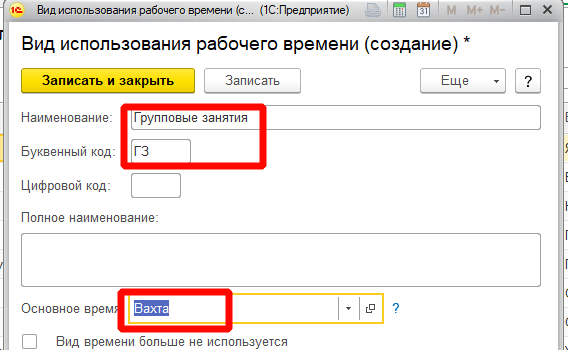


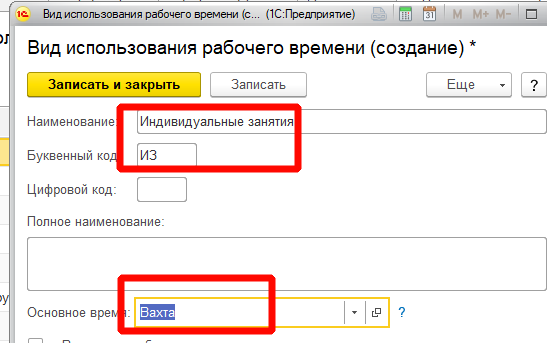




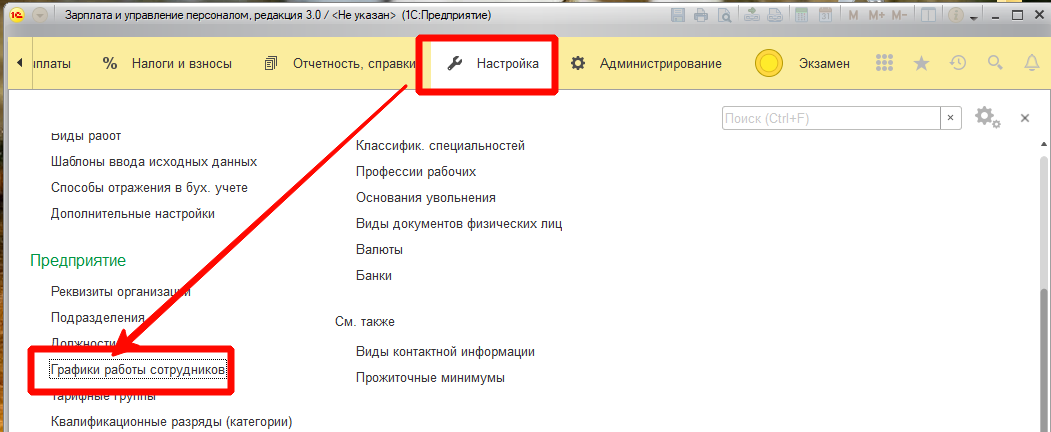
1. **Добавляем два вида рабочего времени:**

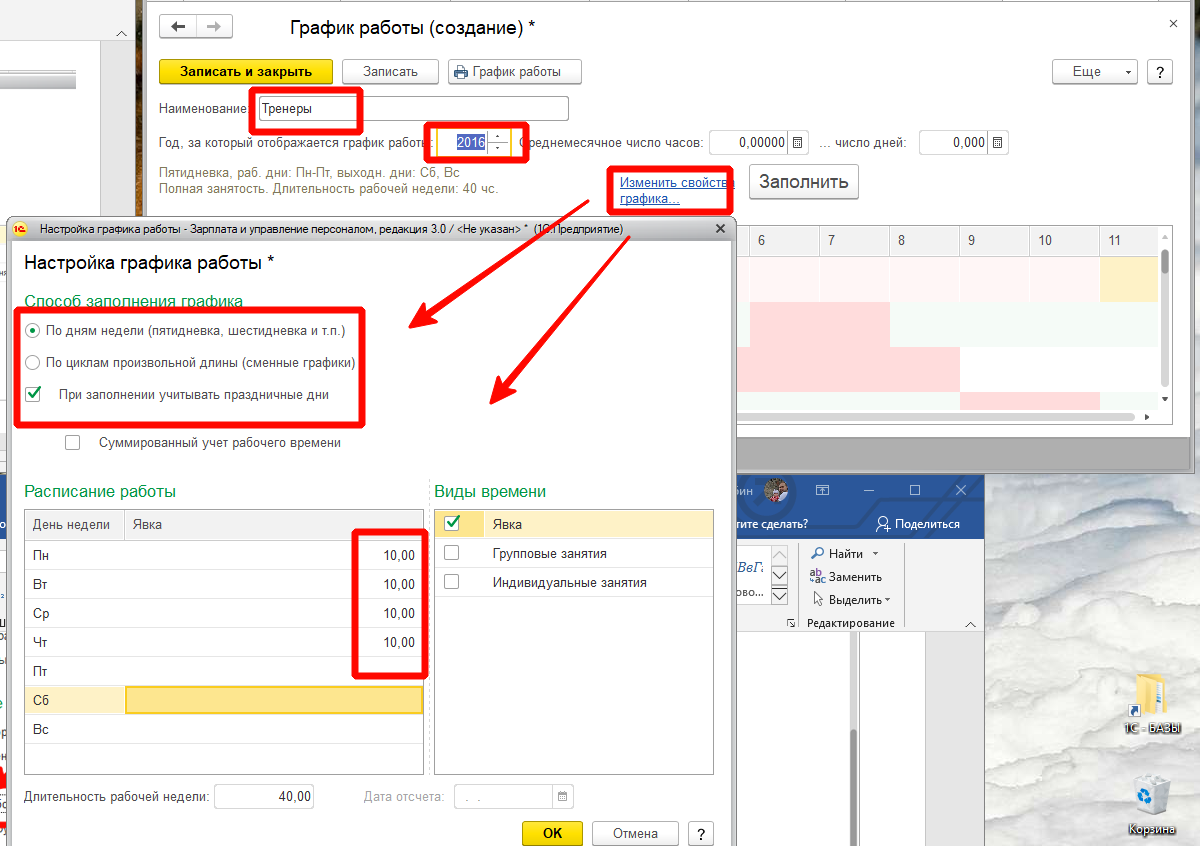




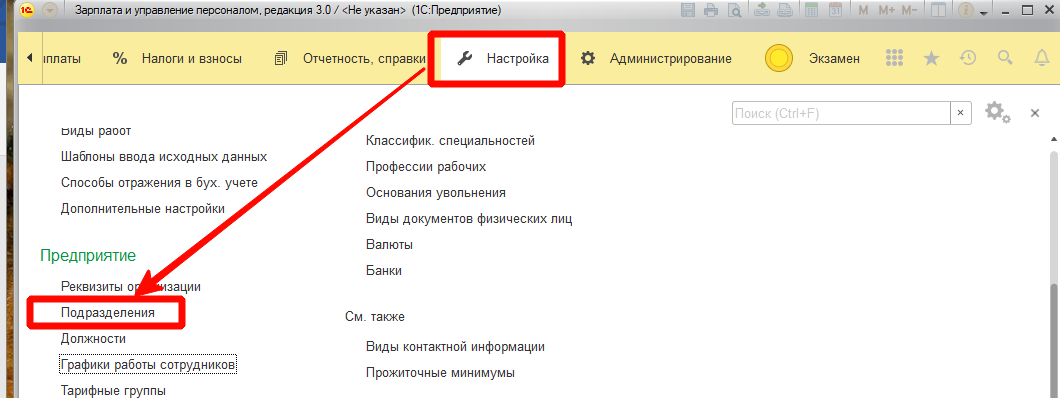


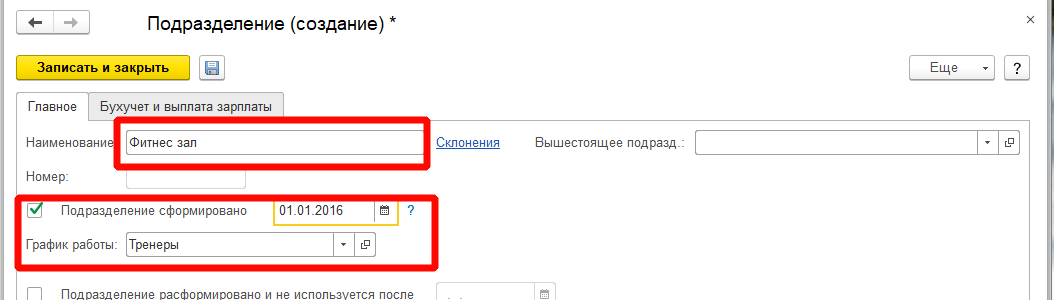
1. **Создаем график работы для Тренеров:**



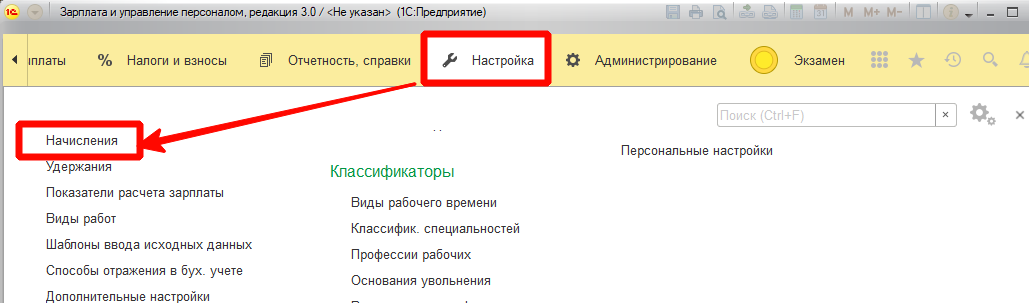


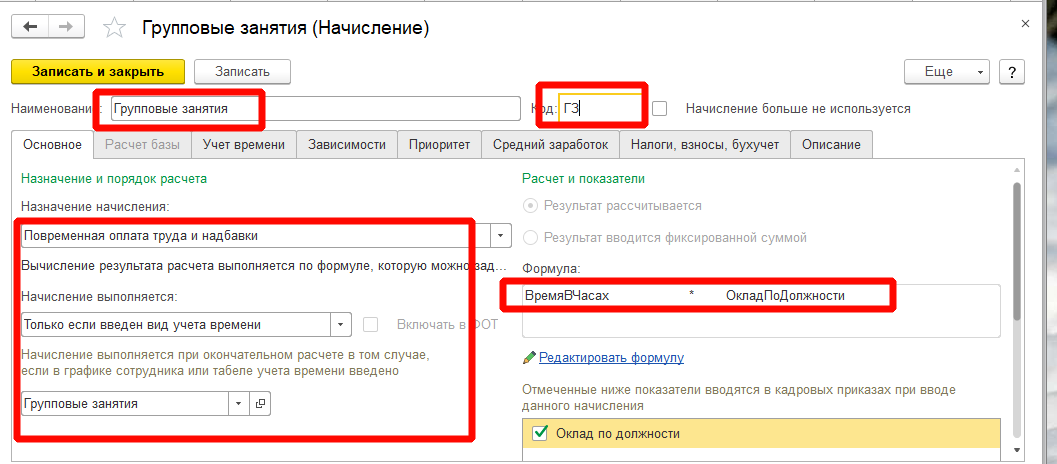
1. **Создаем подразделение:**

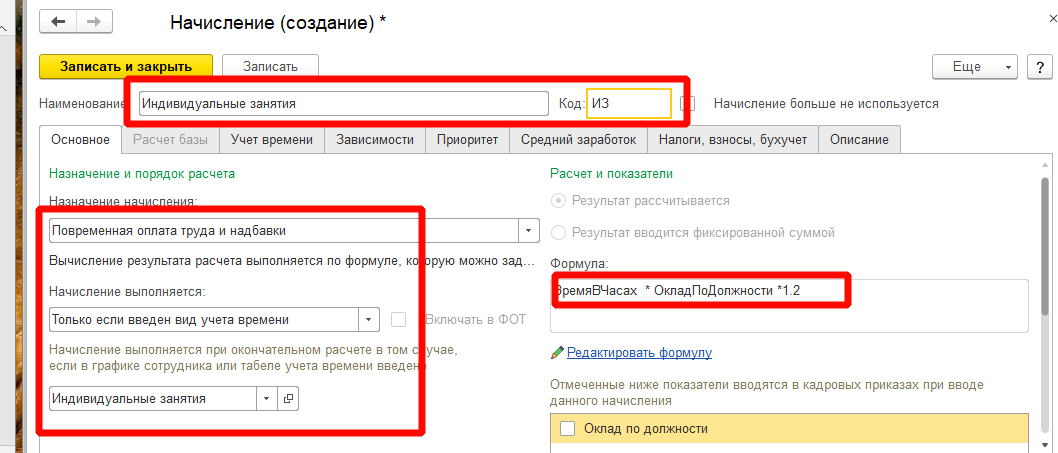




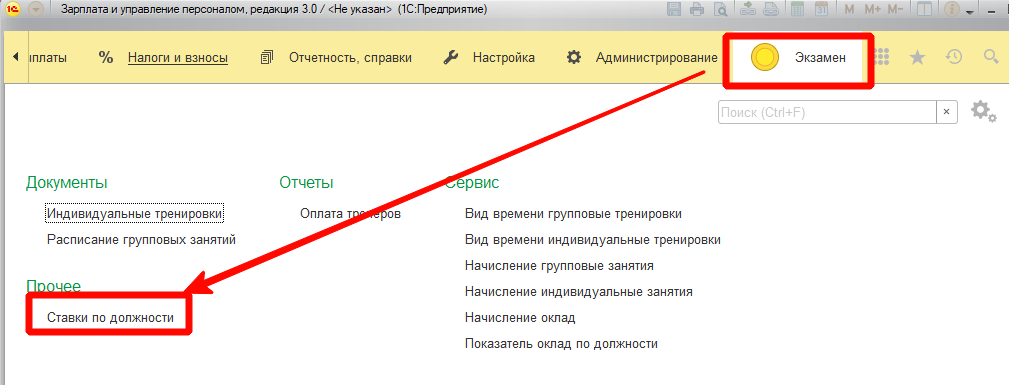
1. **Создаем два начисления:**

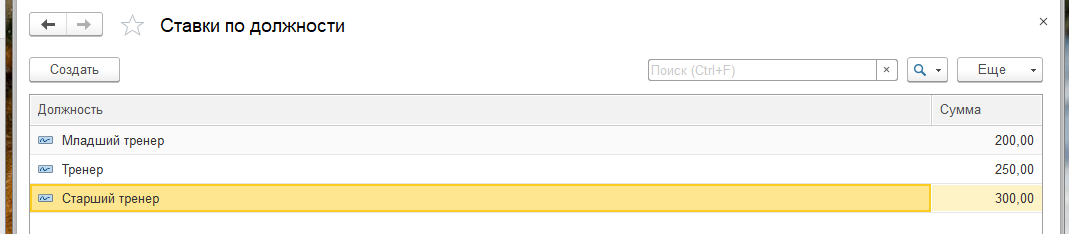




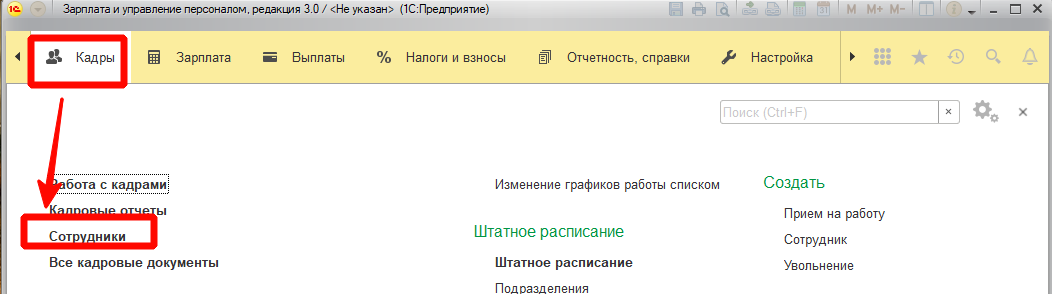


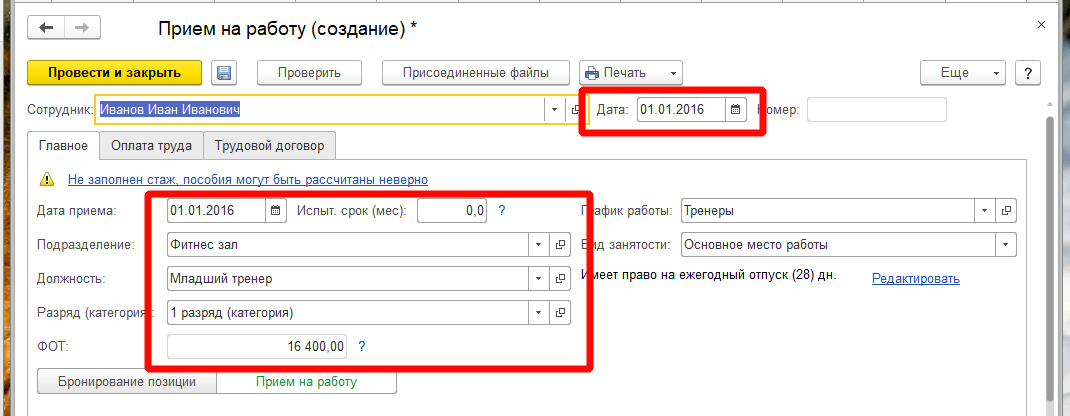
1. **Заполняем ставки по должности:**

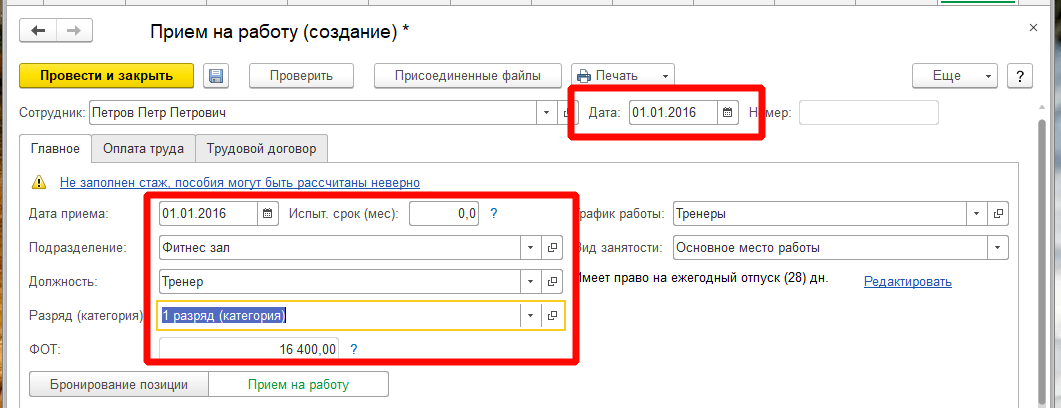




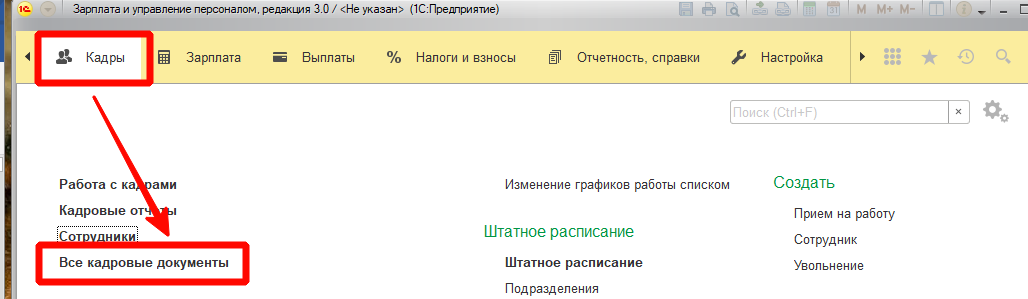
1. **Принимаем на работу Иванова и Петрова 01.01.2016 с оплатой по часовому тарифу:**

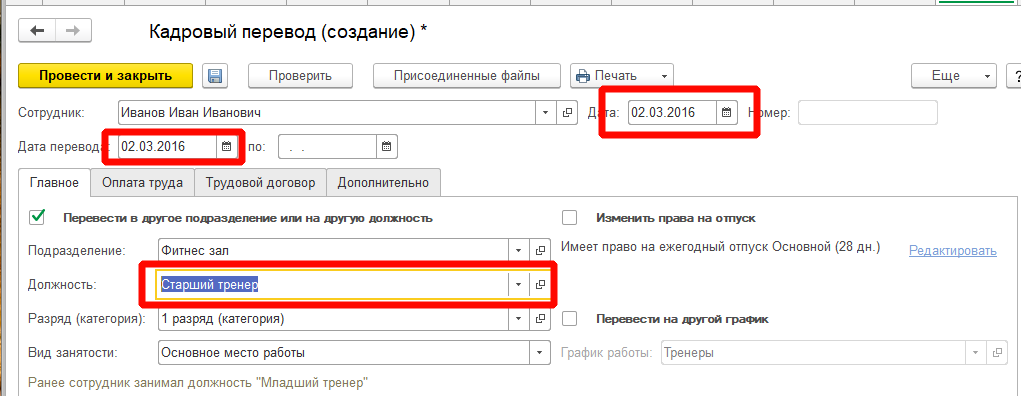




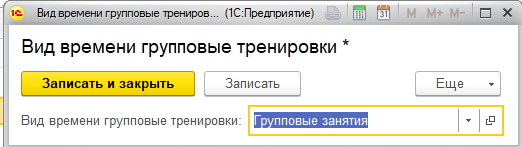


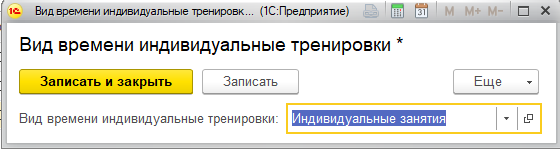
1. **Делаем кадровый перевод:**

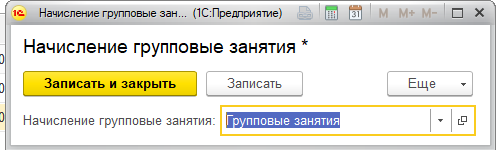


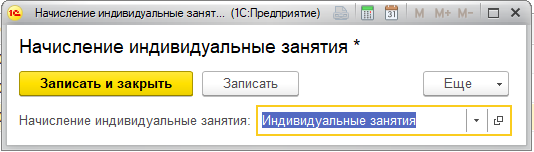


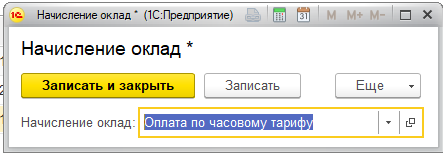
1. **Заполняем константы:**

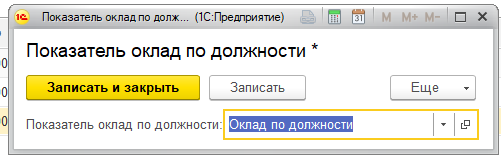




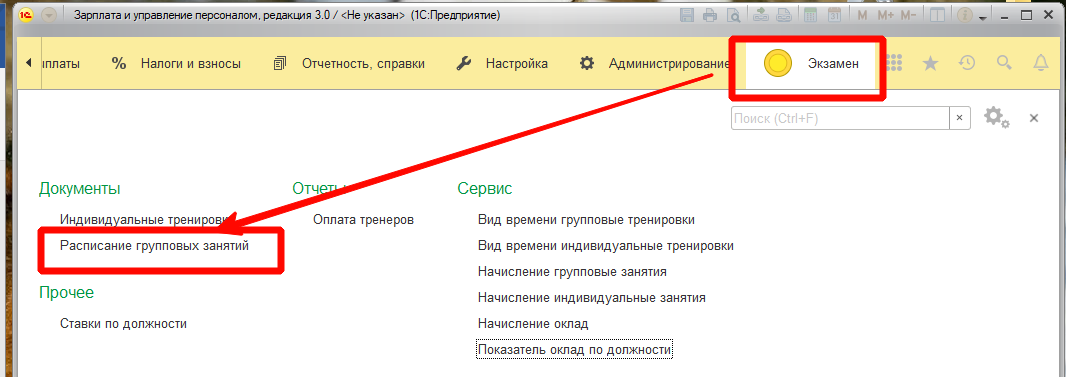


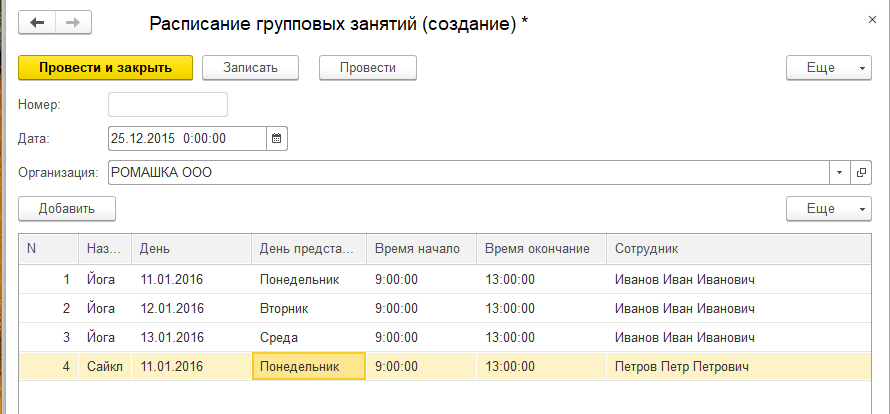


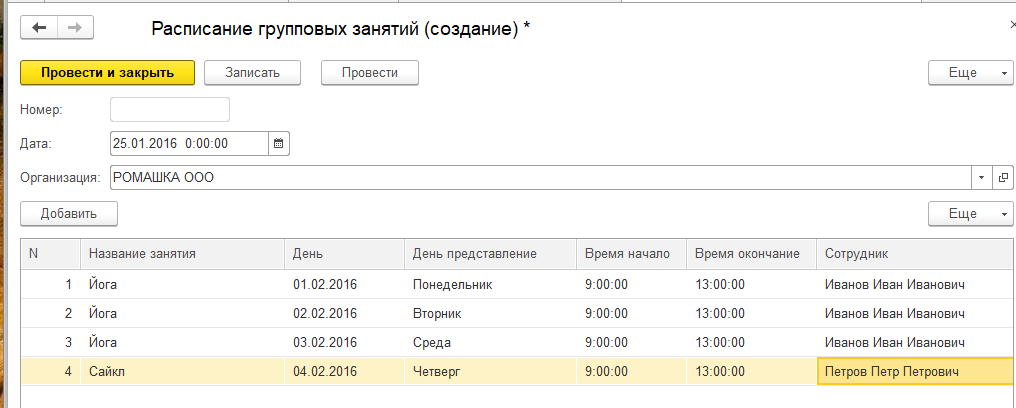


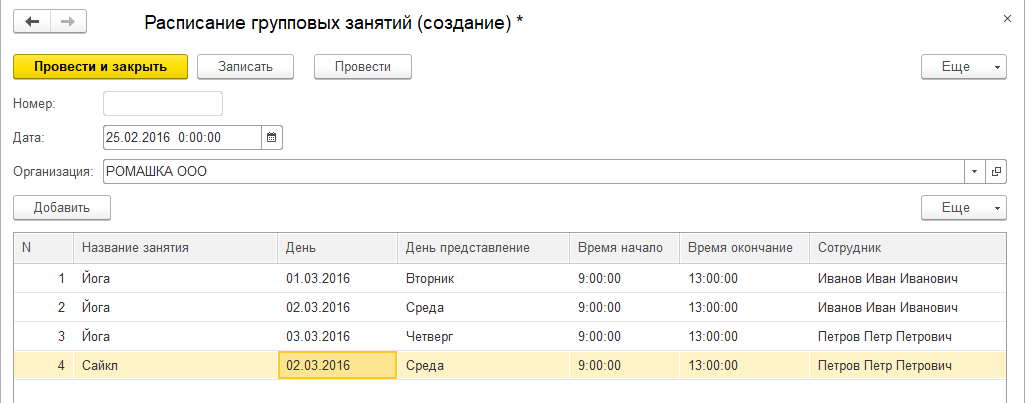


1. **Вводим три документа Расписание групповых занятий:**

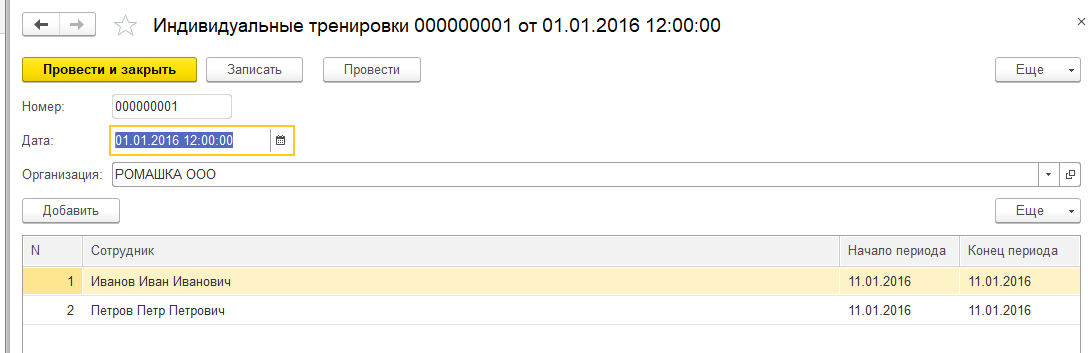




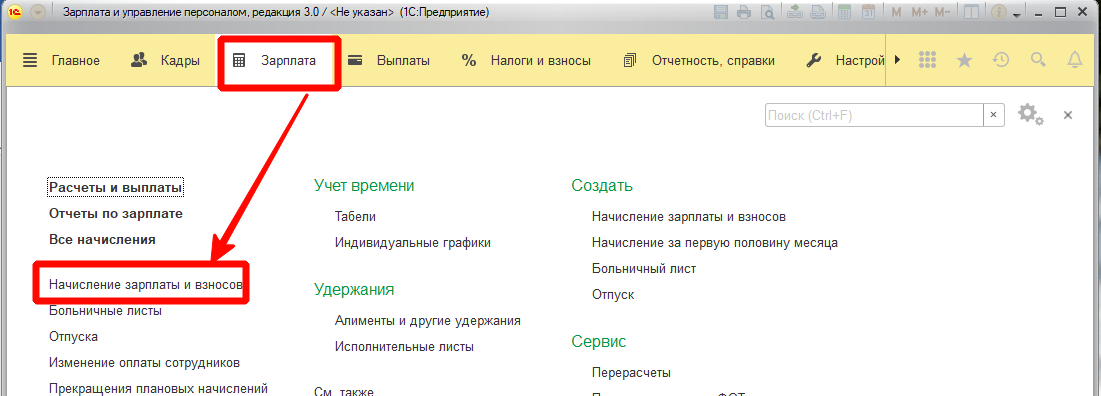


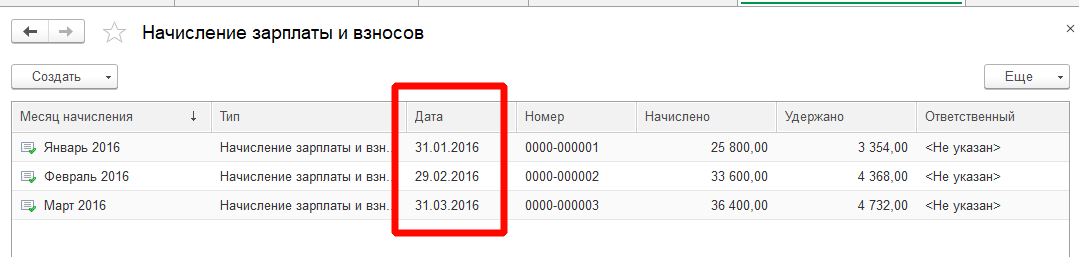


1. **Вводим документ Индивидуальные тренировки:**

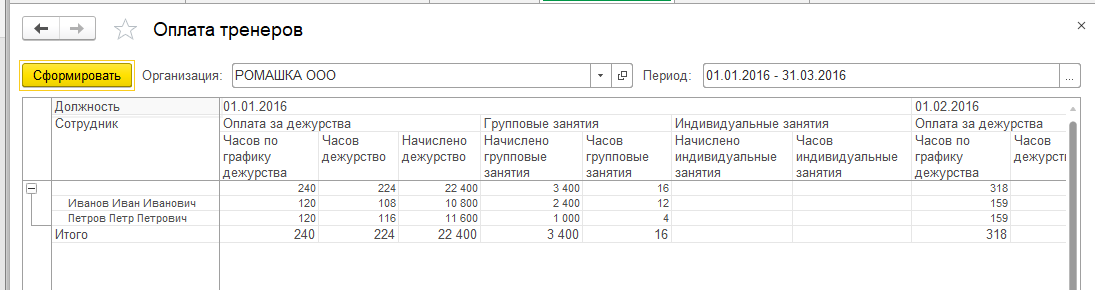


1. **Начисляем ЗП за янв., фев., март 2016:**





**Также смотрим отчет:**



**IV ПОЯСНЕНИЕ К РЕШЕНИЮ**

Для реализации решения созданы 6 констант для хранения показателей расчета зарплаты и начислений.

В соответствии с заданием, для целей автоматизированного начисления работникам оплаты за дежурства и обеспечения учета для целей налогообложения, созданные документы РасписаниеГрупповыхЗанятий и ИндивидуальныеТренировки делают движения в регистры ДанныеОперативногоУчетаРабочегоВремениСотрудников и ЗначенияПериодическихПоказателейРасчетаЗарплатыСотрудников

Создан настраиваемый отчет ОплатаТренеров в соответствии с образцом. Отчет проверен на работоспособность.

Все объекты конфигурации, созданные в процессе решения задачи помещены в подсистему Экзамен.

В базе данных приведен контрольный пример начисления за 3 месяца по 2-м сотрудникам.

Таким образом все требуемые в задании механизмы реализованы.