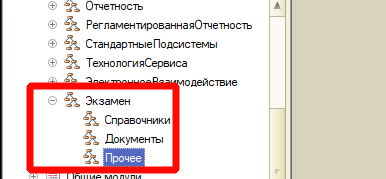
**БИЛЕТ 2**

**РЕШЕНИЕ:**

**I КОНФИГУРАТОР.**

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



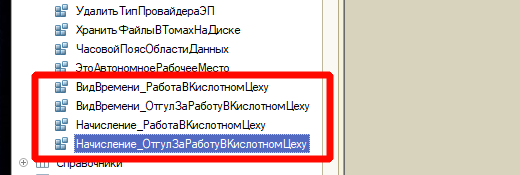
1. **Делаем 4 константы (подсистема Экзамен-Прочее):**

**ВидВремени\_РаботаВКислотномЦеху**, **ВидВремени\_ОтгулЗаРаботуВКислотномЦеху**

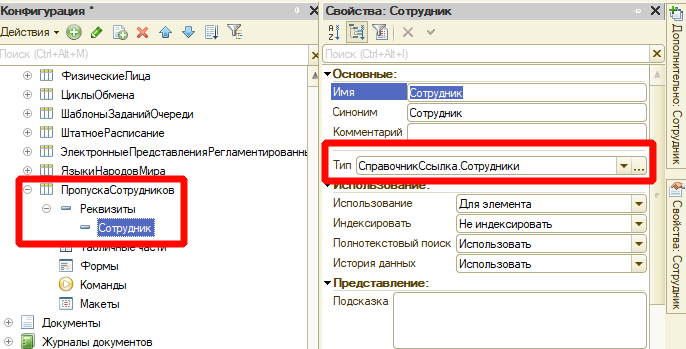
– тип СправочникСсылка.ВидыИспользованияРабочегоВремени

**Начисление\_РаботаВКислотномЦеху**, **Начисление\_ОтгулЗаРаботуВКислотномЦеху**

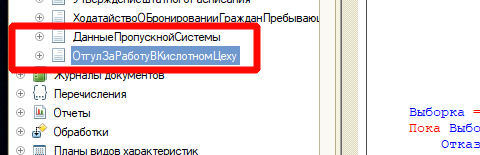
– тип ПланВидовРасчетаСсылка.Начисления



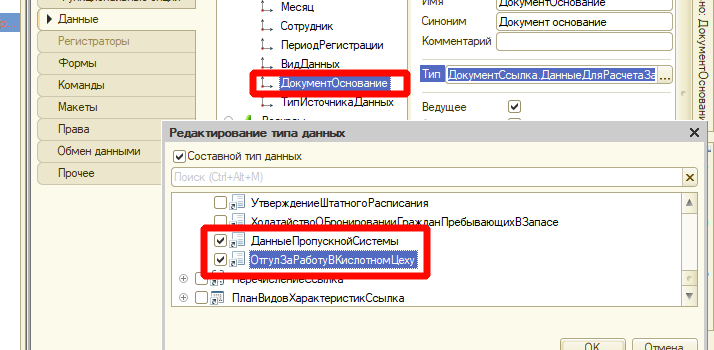
1. **Создаем справочник ПропускаСотрудников (подсистема Экзамен-Справочники):**



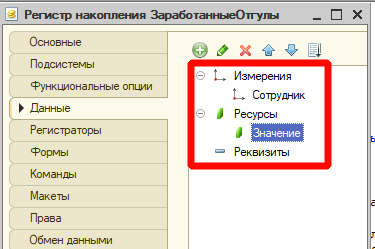
1. **Создаем два документа ДанныеПропускнойСистемы и ОтгулЗаРаботуВКислотномЦеху** **(Экзамен-Документы):**



1. **Ищем регистр сведений ПараметрыЗарегистрированныхДанныхУчетаВремениСотрудников в измерение ДокументОснование добавляем созданные документы:**

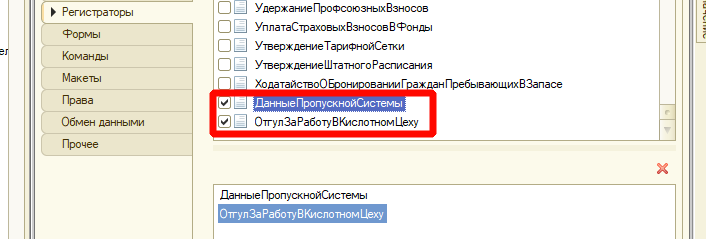


**6. Создаем регистр накопления с типом «остатки» ЗаработанныеОтгулы (подсистема экзамен-прочее):**

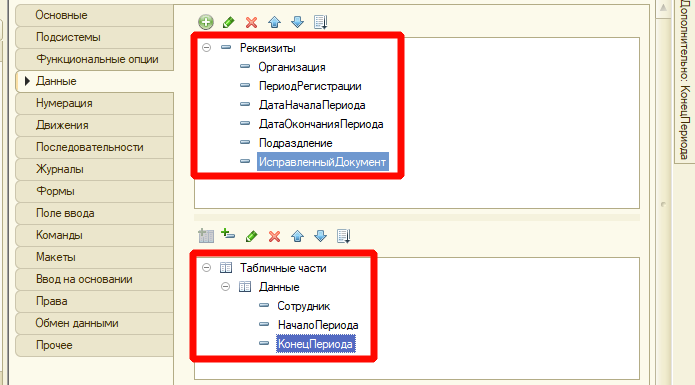


Измерение – **Сотрудник**(СправочникСсылка.Сотрудники),

Ресурс – **Значение**(Число 10.2),



1. **Заполняем документ ДанныеПропускнойСистемы:**



- **Организация** (СправочникСсылка.Организации)

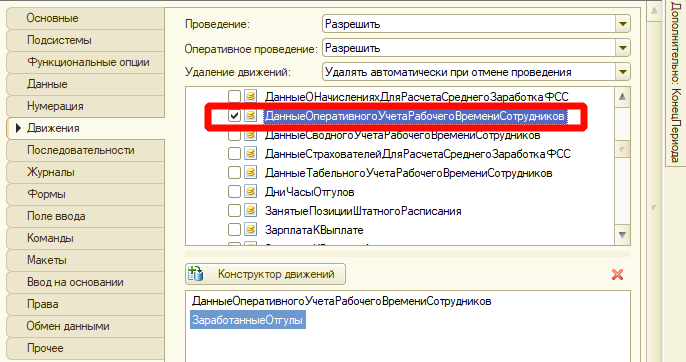
- **ПериодРегистрации**, **ДатаНачалаПериода**, **ДатаОкончанияПериода** – тип Дата

- **Подразделение** – тип СправочникСсылка.ПодразделенияОрганизаций

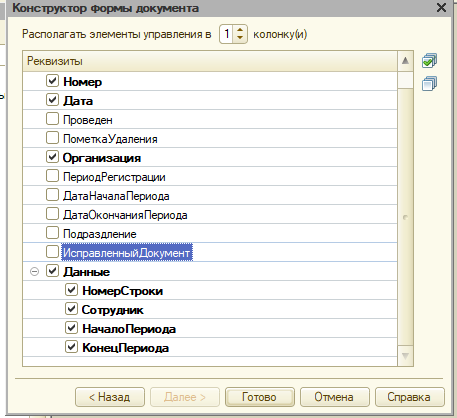
- **ИсправленныйДокумент** – Булево

Табличная часть – **Данные**

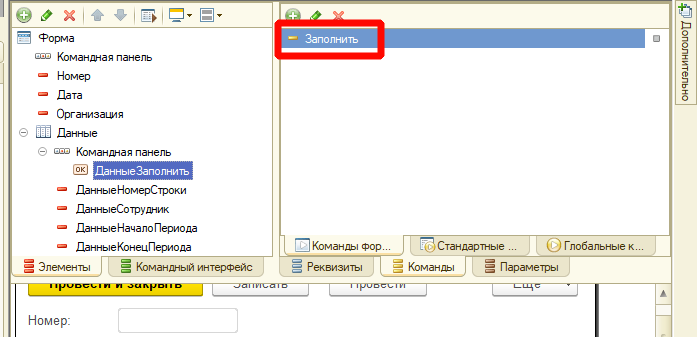
- **Сотрудник** (СправочникСсылка.Сотрудники), **НачалоПериода**, **КонецПериода** – тип Дата(!!!**состав** – Дата и Время)



**Создаем форму этого документа:**

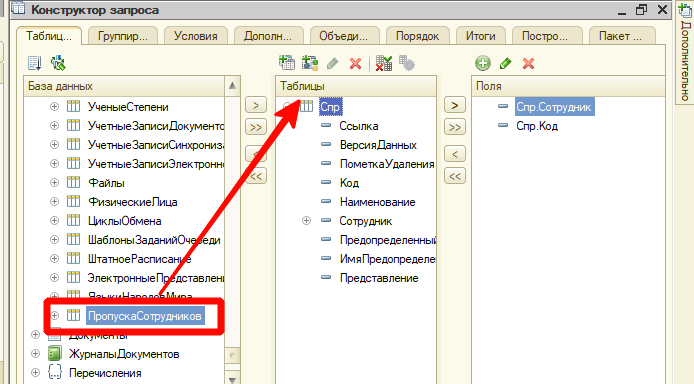


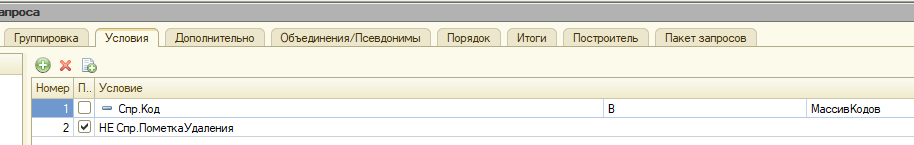
**Добавляем кнопку Заполнить (на клиенте):**



**МОДУЛЬ ФОРМЫ:**

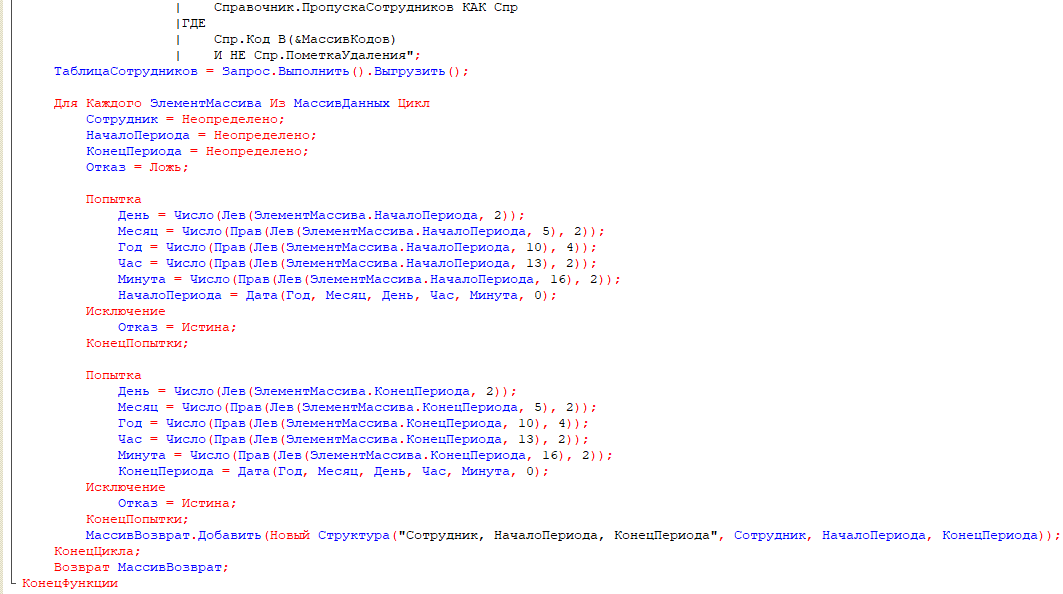
**ЗАПРОС:**





**МОДУЛЬ ПОЛНОСТЬЮ:**







&НаКлиенте

Процедура Заполнить(Команда)

ОписаниеОповещения = Новый ОписаниеОповещения("ПослеВыбораФайла", ЭтаФорма);

ОткрытьФорму("Документ.ДанныеПропускнойСистемы.Форма.ФормаВыбораФайла",, ЭтаФорма,, ВариантОткрытияОкна.ОтдельноеОкно,, ОписаниеОповещения);

КонецПроцедуры

&НаКлиенте

Процедура ПослеВыбораФайла(АдресФайла, ДопПараметр) Экспорт

Попытка

ЧтениеТекста = Новый ЧтениеТекста(АдресФайла);

ТекстФайла = ЧтениеТекста.Прочитать();

Исключение

КонецПопытки;

МассивДанных = СтруктурироватьФайл(ТекстФайла);

МассивДанных = ПолучитьДанныеНаСервере(МассивДанных);

Объект.Данные.Очистить();

Для Каждого ЭлементМассива Из МассивДанных Цикл

СтрокаТаблицыДанные = Объект.Данные.Добавить();

ЗаполнитьЗначенияСвойств(СтрокаТаблицыДанные, ЭлементМассива);

КонецЦикла;

КонецПроцедуры

Функция ПолучитьДанныеНаСервере(МассивДанных);

МассивВозврат = Новый Массив;

МассивПропусков = Новый Массив;

Для Каждого ЭлементМассива Из МассивДанных Цикл

МассивПропусков.Добавить(ЭлементМассива.КодПропуска);

КонецЦикла;

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("МассивКодов", МассивПропусков);

Запрос.Текст = "ВЫБРАТЬ

| Спр.Сотрудник,

| Спр.Код

|ИЗ

| Справочник.ПропускаСотрудников КАК Спр

|ГДЕ

| Спр.Код В(&МассивКодов)

| И НЕ Спр.ПометкаУдаления";

ТаблицаСотрудников = Запрос.Выполнить().Выгрузить();

Для Каждого ЭлементМассива Из МассивДанных Цикл

Сотрудник = Неопределено;

НачалоПериода = Неопределено;

КонецПериода = Неопределено;

Отказ = Ложь;

Попытка

День = Число(Лев(ЭлементМассива.НачалоПериода, 2));

Месяц = Число(Прав(Лев(ЭлементМассива.НачалоПериода, 5), 2));

Год = Число(Прав(Лев(ЭлементМассива.НачалоПериода, 10), 4));

Час = Число(Прав(Лев(ЭлементМассива.НачалоПериода, 13), 2));

Минута = Число(Прав(Лев(ЭлементМассива.НачалоПериода, 16), 2));

НачалоПериода = Дата(Год, Месяц, День, Час, Минута, 0);

Исключение

Отказ = Истина;

КонецПопытки;

Попытка

День = Число(Лев(ЭлементМассива.КонецПериода, 2));

Месяц = Число(Прав(Лев(ЭлементМассива.КонецПериода, 5), 2));

Год = Число(Прав(Лев(ЭлементМассива.КонецПериода, 10), 4));

Час = Число(Прав(Лев(ЭлементМассива.КонецПериода, 13), 2));

Минута = Число(Прав(Лев(ЭлементМассива.КонецПериода, 16), 2));

КонецПериода = Дата(Год, Месяц, День, Час, Минута, 0);

Исключение

Отказ = Истина;

КонецПопытки;

МассивВозврат.Добавить(Новый Структура("Сотрудник, НачалоПериода, КонецПериода", Сотрудник, НачалоПериода, КонецПериода));

КонецЦикла;

Возврат МассивВозврат;

КонецФункции

&НаКлиенте

Функция СтруктурироватьФайл(ТекстФайла)

МассивВозврат = Новый Массив;

Для ИИ = 1 По СтрЧислоСтрок(ТекстФайла) Цикл

КодПропуска = "";

НачалоПериода = "";

КонецПериода = "";

Строка = СтрПолучитьСтроку(ТекстФайла, ИИ);

Если СокрЛП(Строка) = "" Тогда

Продолжить;

КонецЕсли;

ПозицияРазделителя = СтрНайти(Строка, ";");

Если ПозицияРазделителя > 0 Тогда

КодПропуска = Лев(Строка, ПозицияРазделителя - 1);

Строка = Прав(Строка, СтрДлина(Строка) - ПозицияРазделителя);

Иначе

КодПропуска = Строка;

Строка = "";

КонецЕсли;

КонецПериода = Строка;

МассивВозврат.Добавить(Новый Структура("КодПропуска, НачалоПериода, КонецПериода", СокрЛП(КодПропуска), СокрЛП(НачалоПериода), СокрЛП(КонецПериода)));

КонецЦикла;

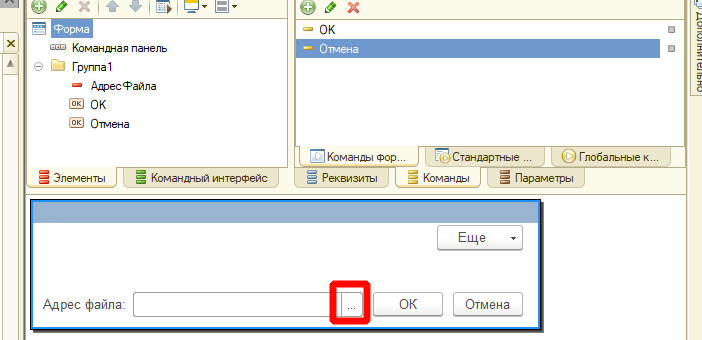
Возврат МассивВозврат;

КонецФункции

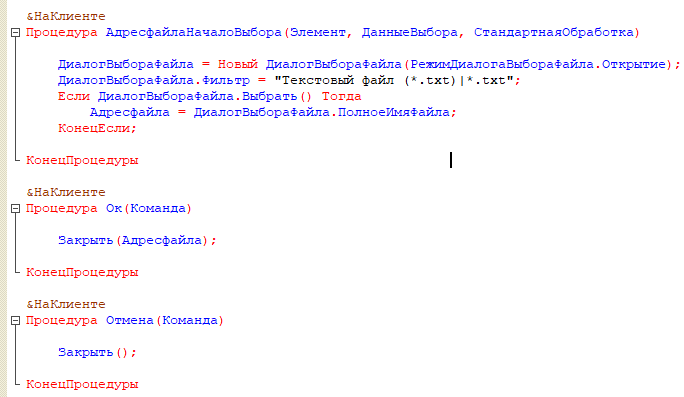
**Сюда же добавляем форму выбора файла (произвольная форма):**

**Добавляем АдресФайла с кнопкой выбора, событие – НачалоВыбора – на клиенте**

**Кнопки ОК и Отмена – тоже не клиенте**



**Модуль формы выбора:**



&НаКлиенте

Процедура АдресфайлаНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

ДиалогВыбораФайла = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

ДиалогВыбораФайла.Фильтр = "Текстовый файл (\*.txt)|\*.txt";

Если ДиалогВыбораФайла.Выбрать() Тогда

Адресфайла = ДиалогВыбораФайла.ПолноеИмяФайла;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура Ок(Команда)

Закрыть(Адресфайла);

КонецПроцедуры

&НаКлиенте

Процедура Отмена(Команда)

Закрыть();

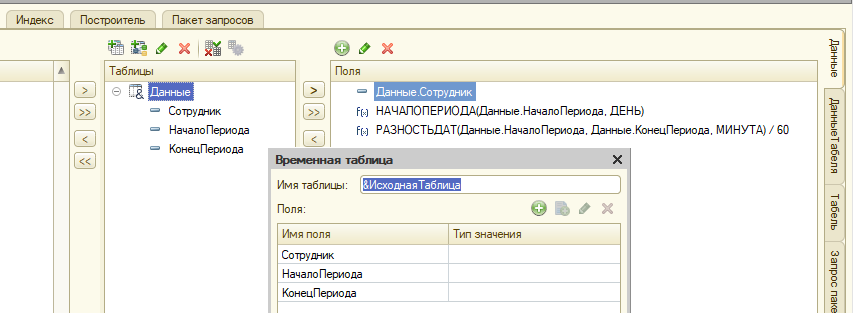
КонецПроцедуры

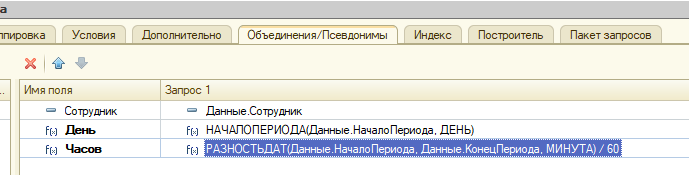
**Модуль документа ДанныеПропускнойСистемы:**

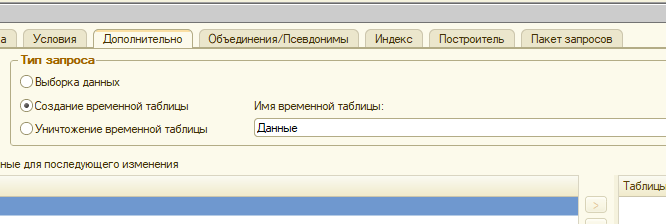
**ЗАПРОС:**

**ПЕРВЫЙ ПАКЕТ:**

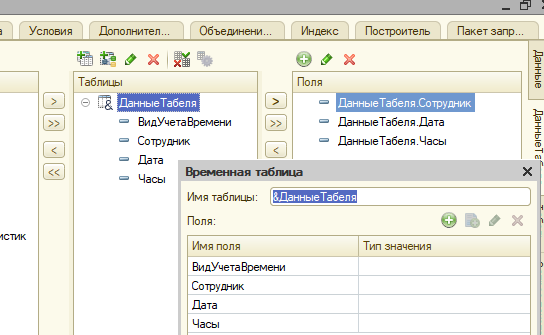
**если выдает ошибку, прописывать потом вручную:**



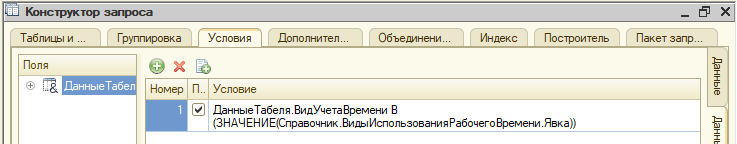


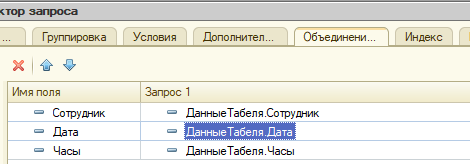


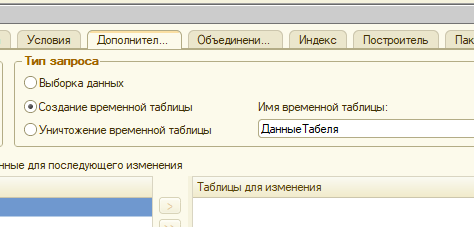
**ВТОРОЙ ПАКЕТ:**



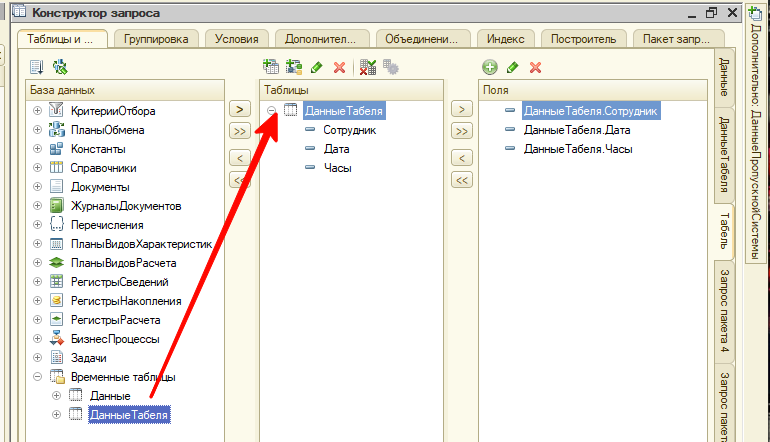
**Если выдает ошибку – прописывать потом вручную:**

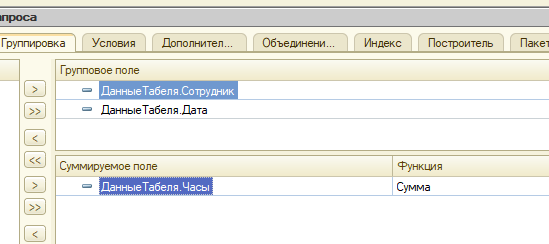


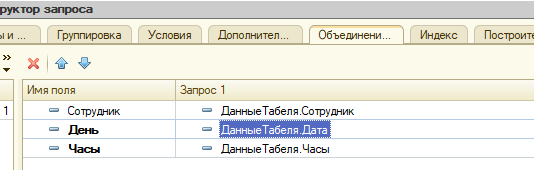


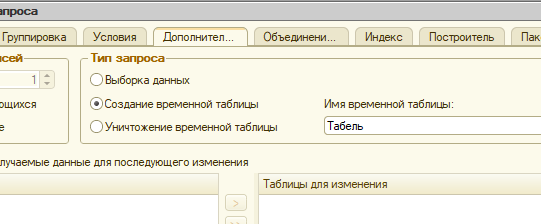


**ТРЕТИЙ ПАКЕТ:**

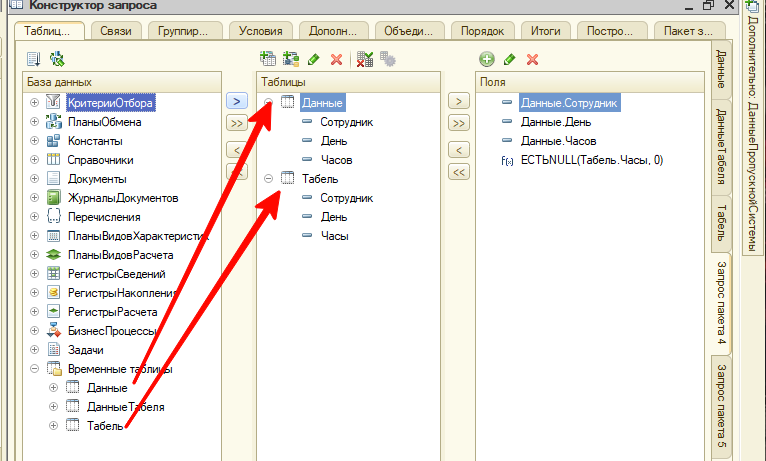


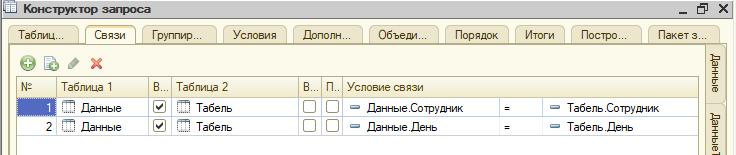


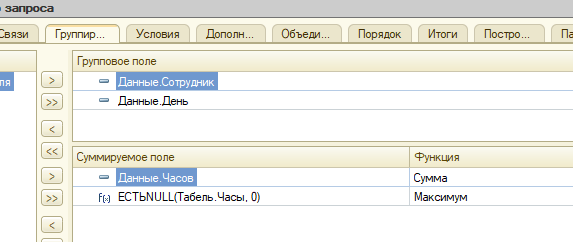


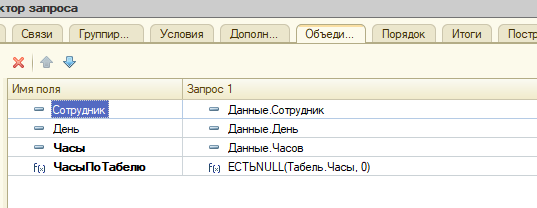


**ЧЕТВЕРТЫЙ ПАКЕТ:**

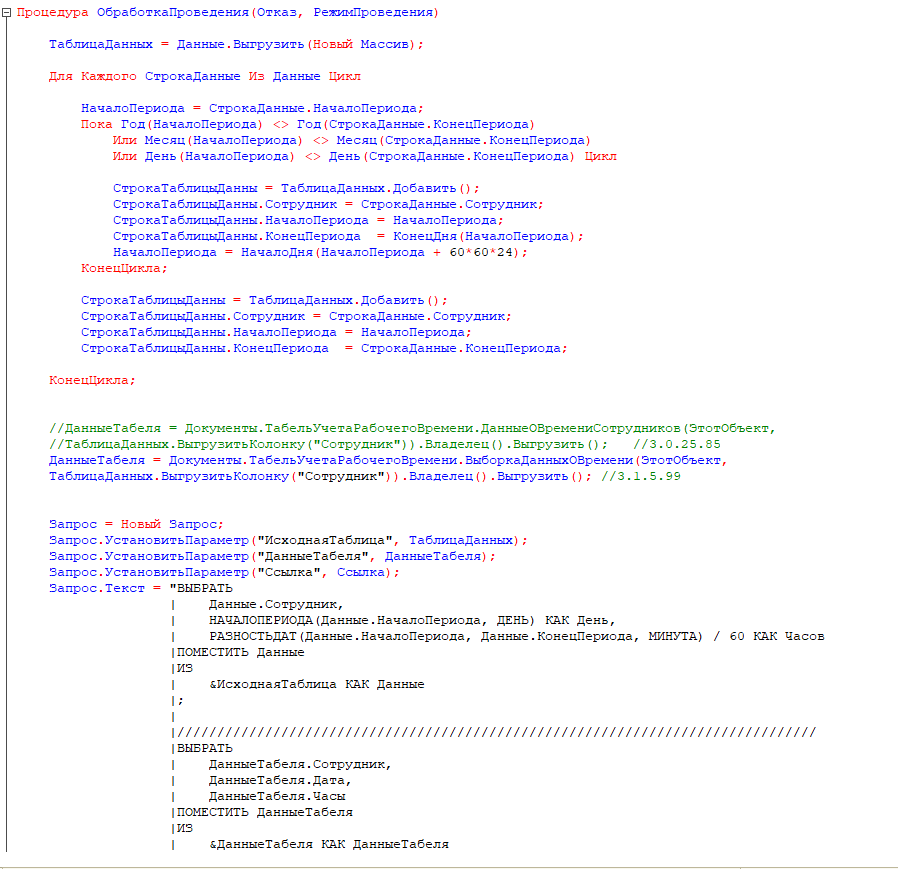




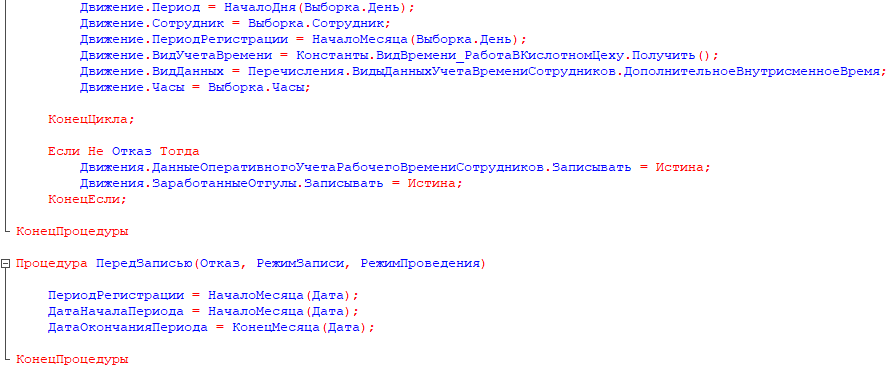




**Модуль полностью:**







Процедура ОбработкаПроведения(Отказ, РежимПроведения)

ТаблицаДанных = 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

СтрокаТаблицыДанны = ТаблицаДанных.Добавить();

СтрокаТаблицыДанны.Сотрудник = СтрокаДанные.Сотрудник;

СтрокаТаблицыДанны.НачалоПериода = НачалоПериода;

СтрокаТаблицыДанны.КонецПериода = КонецДня(НачалоПериода);

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

КонецЦикла;

СтрокаТаблицыДанны = ТаблицаДанных.Добавить();

СтрокаТаблицыДанны.Сотрудник = СтрокаДанные.Сотрудник;

СтрокаТаблицыДанны.НачалоПериода = НачалоПериода;

СтрокаТаблицыДанны.КонецПериода = СтрокаДанные.КонецПериода;

КонецЦикла;

//ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ДанныеОВремениСотрудников(ЭтотОбъект,

//ТаблицаДанных.ВыгрузитьКолонку("Сотрудник")).Владелец().Выгрузить(); //3.0.25.85

ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ВыборкаДанныхОВремени(ЭтотОбъект,

ТаблицаДанных.ВыгрузитьКолонку("Сотрудник")).Владелец().Выгрузить(); //3.1.5.99

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("ИсходнаяТаблица", ТаблицаДанных);

Запрос.УстановитьПараметр("ДанныеТабеля", ДанныеТабеля);

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.Текст = "ВЫБРАТЬ

| Данные.Сотрудник,

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

| РАЗНОСТЬДАТ(Данные.НачалоПериода, Данные.КонецПериода, МИНУТА) / 60 КАК Часов

|ПОМЕСТИТЬ Данные

|ИЗ

| &ИсходнаяТаблица КАК Данные

|;

|

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

|ВЫБРАТЬ

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата,

| ДанныеТабеля.Часы

|ПОМЕСТИТЬ ДанныеТабеля

|ИЗ

| &ДанныеТабеля КАК ДанныеТабеля

|ГДЕ

| ДанныеТабеля.ВидУчетаВремени В (ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка))

|;

|

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

|ВЫБРАТЬ

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата КАК День,

| СУММА(ДанныеТабеля.Часы) КАК Часы

|ПОМЕСТИТЬ Табель

|ИЗ

| ДанныеТабеля КАК ДанныеТабеля

|

|СГРУППИРОВАТЬ ПО

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата

|;

|

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

|ВЫБРАТЬ

| Данные.Сотрудник,

| Данные.День,

| СУММА(Данные.Часов) КАК Часы,

| МАКСИМУМ(ЕСТЬNULL(Табель.Часы, 0)) КАК ЧасыПоТабелю

|ИЗ

| Данные КАК Данные

| ЛЕВОЕ СОЕДИНЕНИЕ Табель КАК Табель

| ПО Данные.Сотрудник = Табель.Сотрудник

| И Данные.День = Табель.День

|

|СГРУППИРОВАТЬ ПО

| Данные.Сотрудник,

| Данные.День

|";

Выборка= Запрос.Выполнить().Выбрать();

Пока Выборка.Следующий() Цикл

Если Выборка.Часы > Выборка.ЧасыПоТабелю Тогда

СообщениеПользователю = Новый СообщениеПользователю;

СообщениеПользователю.Текст = "Ошибка проведения. Проверить введенные данные";

СообщениеПользователю.Сообщить();

Отказ = Истина;

КонецЕсли;

Движение = Движения.ЗаработанныеОтгулы.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Сотрудник = Выборка.Сотрудник;

Движение.Значение = Выборка.Часы;

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = НачалоДня(Выборка.День);

Движение.Сотрудник = Выборка.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(Выборка.День);

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

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДополнительноеВнутрисменноеВремя;

Движение.Часы = Выборка.Часы;

КонецЦикла;

Если Не Отказ Тогда

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ЗаработанныеОтгулы.Записывать = Истина;

КонецЕсли;

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

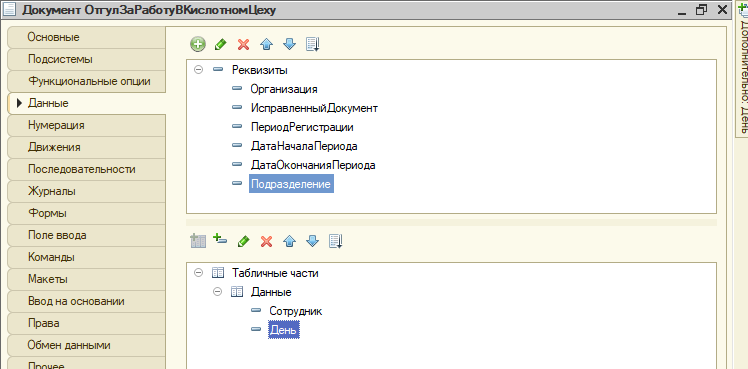
ПериодРегистрации = НачалоМесяца(Дата);

ДатаНачалаПериода = НачалоМесяца(Дата);

ДатаОкончанияПериода = КонецМесяца(Дата);

КонецПроцедуры

1. **Возвращаемся к документу «ОтгулЗаРаботуВКислотномЦеху»:**



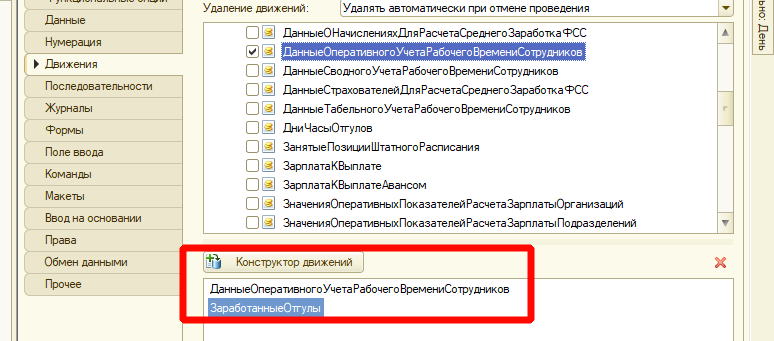
**Организация** - СправочникСсылка.Организации,

**ИсправленныйДокумент** –Булево,

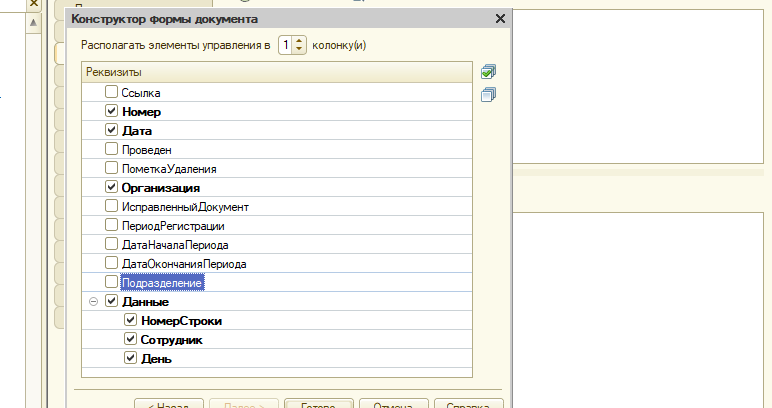
**ПериодРегистрации**, **ДатаНачалаПериода**, **ДатаОкончанияПериода** – Дата,

**Подразделение** - СправочникСсылка.ПодразделенияОрганизаций

Табличная часть – **Данные** - **Сотрудник** (СправочникСсылка.Сотрудники), **День** – Дата



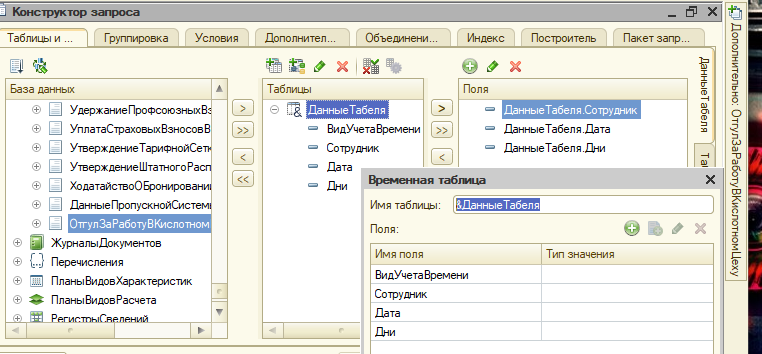
**Делаем форму документа без модуля:**



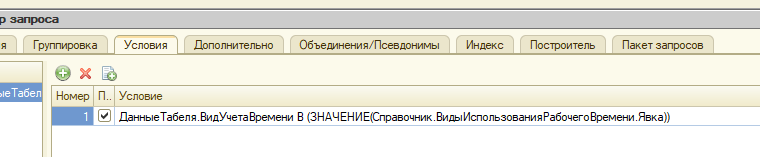
**Модуль документа ОтгулЗаРаботуВКислотномЦеху:**

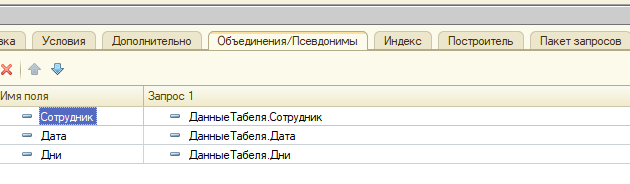
**ЗАПРОС:**

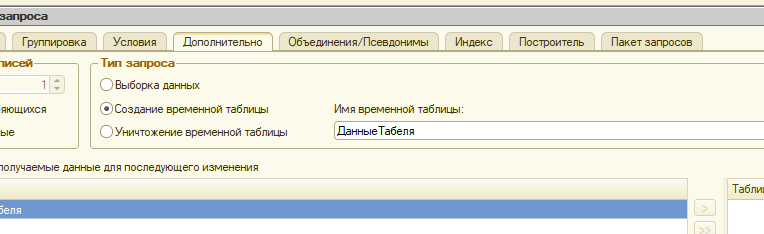
**ПЕРВЫЙ ПАКЕТ:**



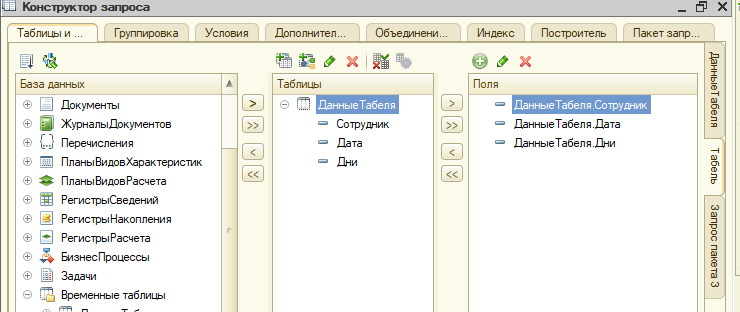
**Если выдает ошибку – прописывать потом вручную:**

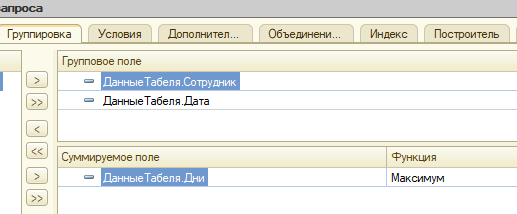


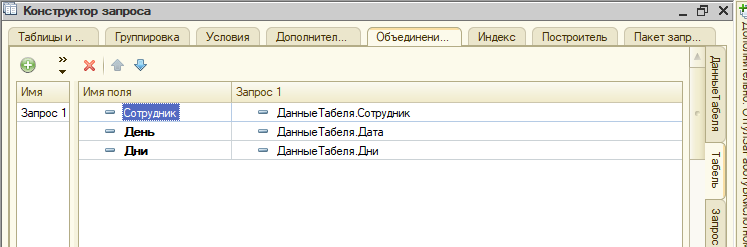


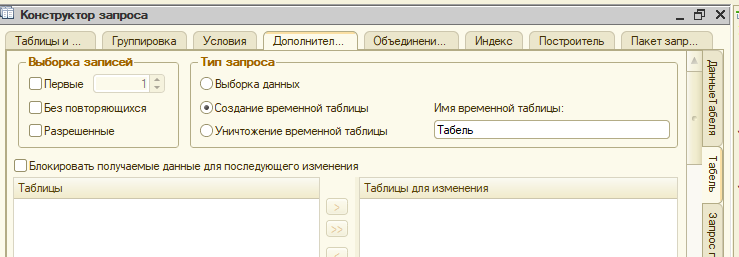


**ВТОРОЙ ПАКЕТ:**

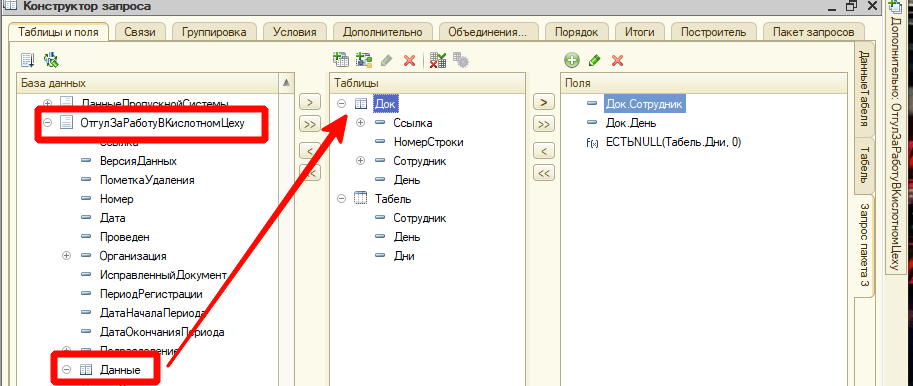


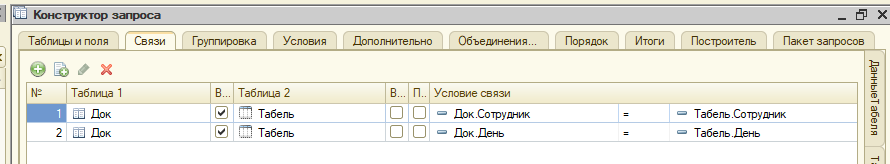


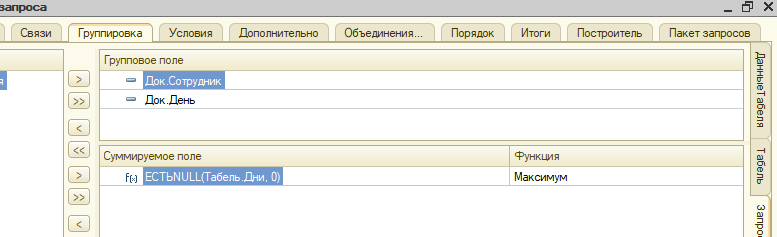


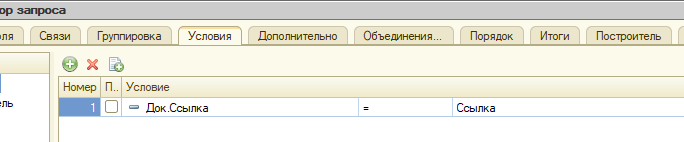


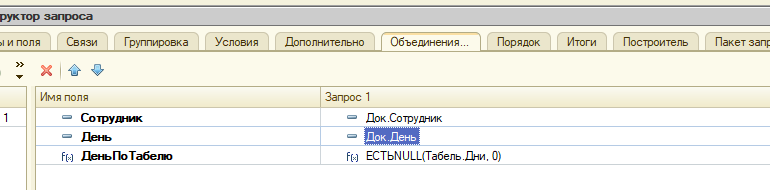
**ТРЕТИЙ ПАКЕТ:**

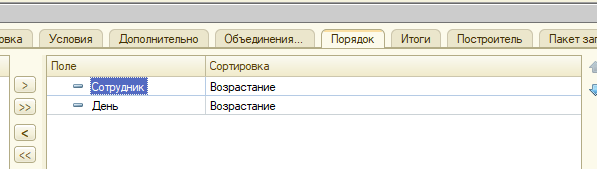


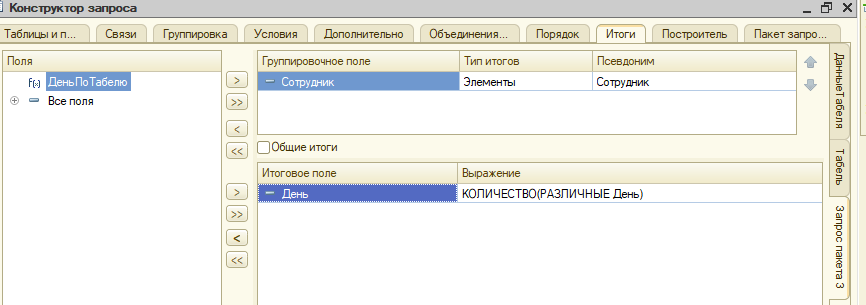




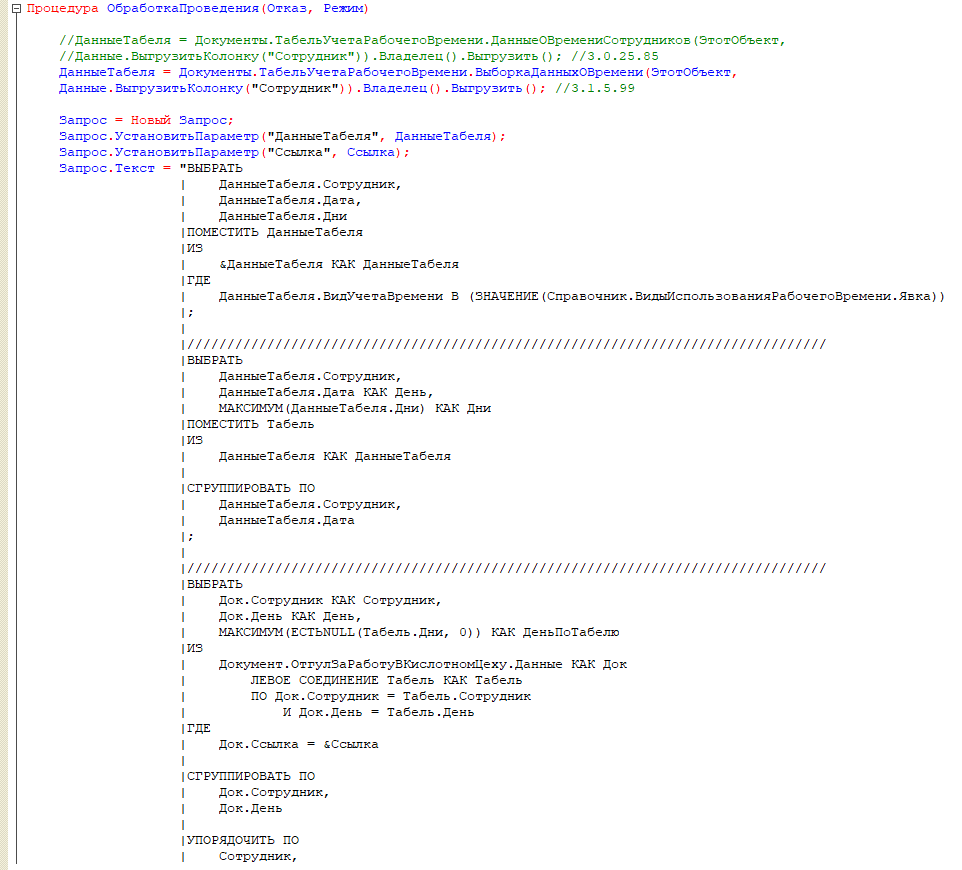




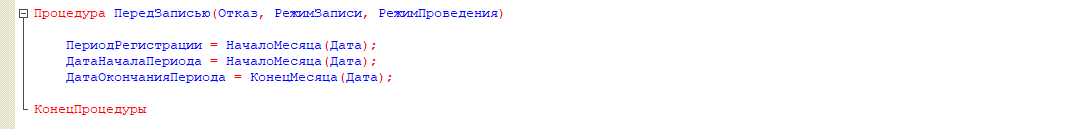




**МОДУЛЬ документа ОтгулЗаРаботуВКислотномЦеху полностью:**







Процедура ОбработкаПроведения(Отказ, Режим)

//ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ДанныеОВремениСотрудников(ЭтотОбъект,

//Данные.ВыгрузитьКолонку("Сотрудник")).Владелец().Выгрузить(); //3.0.25.85

ДанныеТабеля = Документы.ТабельУчетаРабочегоВремени.ВыборкаДанныхОВремени(ЭтотОбъект,

Данные.ВыгрузитьКолонку("Сотрудник")).Владелец().Выгрузить(); //3.1.5.99

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("ДанныеТабеля", ДанныеТабеля);

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.Текст = "ВЫБРАТЬ

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата,

| ДанныеТабеля.Дни

|ПОМЕСТИТЬ ДанныеТабеля

|ИЗ

| &ДанныеТабеля КАК ДанныеТабеля

|ГДЕ

| ДанныеТабеля.ВидУчетаВремени В (ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка))

|;

|

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

|ВЫБРАТЬ

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата КАК День,

| МАКСИМУМ(ДанныеТабеля.Дни) КАК Дни

|ПОМЕСТИТЬ Табель

|ИЗ

| ДанныеТабеля КАК ДанныеТабеля

|

|СГРУППИРОВАТЬ ПО

| ДанныеТабеля.Сотрудник,

| ДанныеТабеля.Дата

|;

|

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

|ВЫБРАТЬ

| Док.Сотрудник КАК Сотрудник,

| Док.День КАК День,

| МАКСИМУМ(ЕСТЬNULL(Табель.Дни, 0)) КАК ДеньПоТабелю

|ИЗ

| Документ.ОтгулЗаРаботуВКислотномЦеху.Данные КАК Док

| ЛЕВОЕ СОЕДИНЕНИЕ Табель КАК Табель

| ПО Док.Сотрудник = Табель.Сотрудник

| И Док.День = Табель.День

|ГДЕ

| Док.Ссылка = &Ссылка

|

|СГРУППИРОВАТЬ ПО

| Док.Сотрудник,

| Док.День

|

|УПОРЯДОЧИТЬ ПО

| Сотрудник,

| День

|ИТОГИ

| КОЛИЧЕСТВО(РАЗЛИЧНЫЕ День)

|ПО

| Сотрудник";

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

Пока ВыборкаСотрудник.Следующий() Цикл

Движение = Движения.ЗаработанныеОтгулы.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Сотрудник = ВыборкаСотрудник.Сотрудник;

Движение.Период = Дата;

Движение.Значение = ВыборкаСотрудник.День \* 4;

Выборка = ВыборкаСотрудник.Выбрать();

Пока Выборка.Следующий() Цикл

Если Выборка.ДеньПоТабелю = 0 Тогда

СообщениеПользователю = Новый СообщениеПользователю;

СообщениеПользователю.Текст = "Сотрудник: " + Строка(Выборка.Сотрудник) + " " +

Формат(Выборка.День, "ДЛФ=DD") + " не работает ";

СообщениеПользователю.Сообщить();

Отказ = Истина;

Продолжить;

КонецЕсли;

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = Выборка.День;

Движение.Сотрудник = ВыборкаСотрудник.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(Выборка.День);

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

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДанныеОперативногоУчета;

Движение.Дни = 1;

Движение.Часы = 1;

КонецЦикла;

КонецЦикла;

Если Не Отказ Тогда

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ЗаработанныеОтгулы.Записывать = Истина;

КонецЕсли;

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

ПериодРегистрации = НачалоМесяца(Дата);

ДатаНачалаПериода = НачалоМесяца(Дата);

ДатаОкончанияПериода = КонецМесяца(Дата);

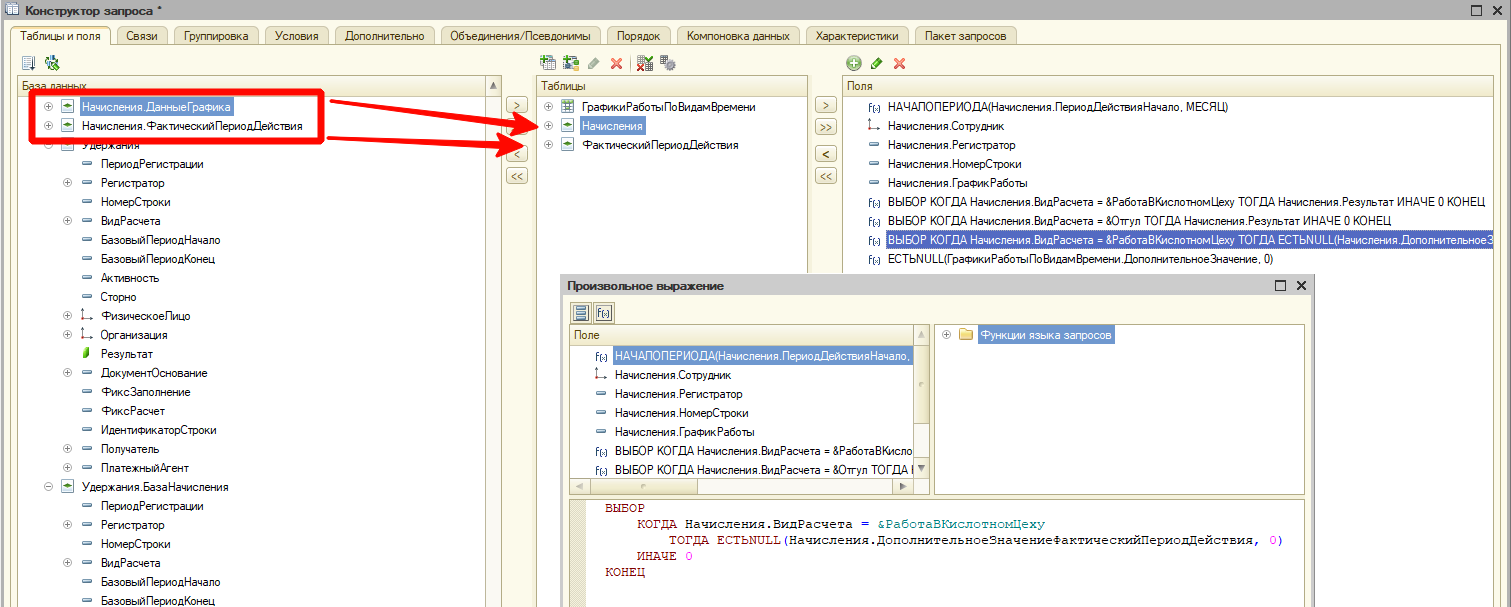
КонецПроцедуры

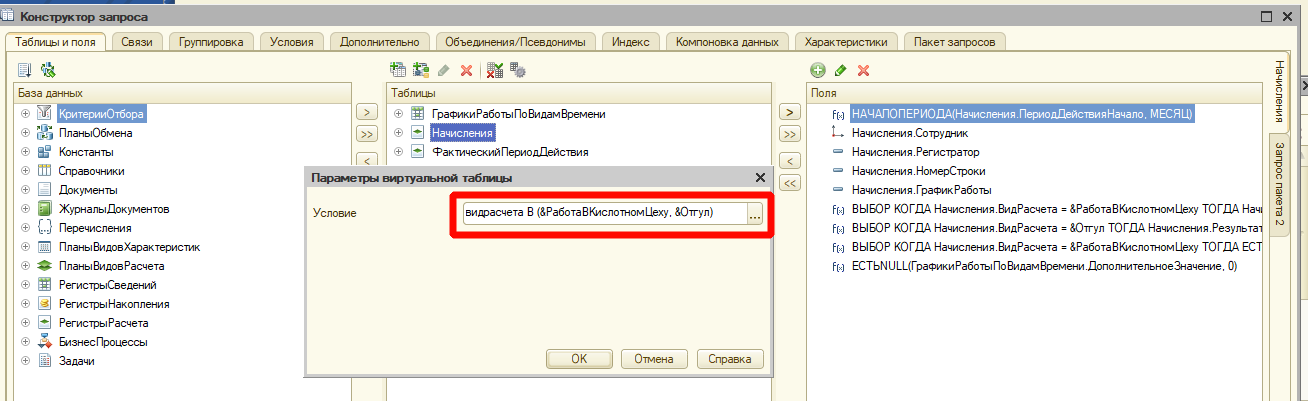
1. **Создаем отчет РаботаВКислотномЦеху (подсистема экзамен):**

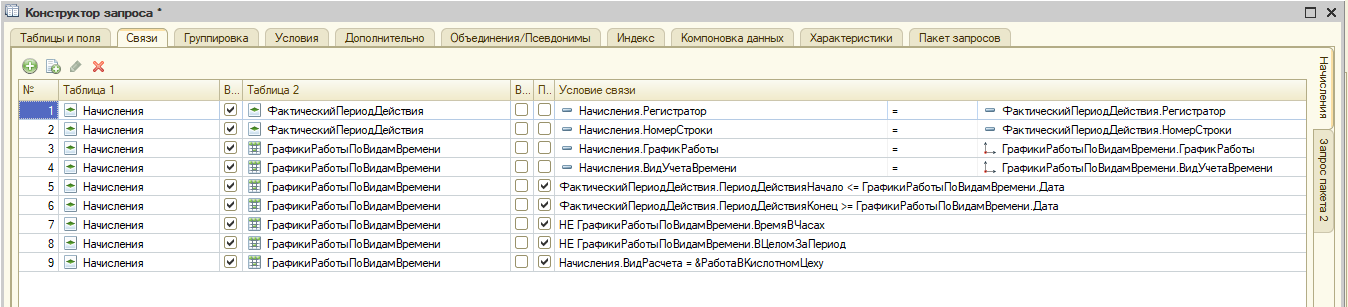
**ЗАПРОС:**

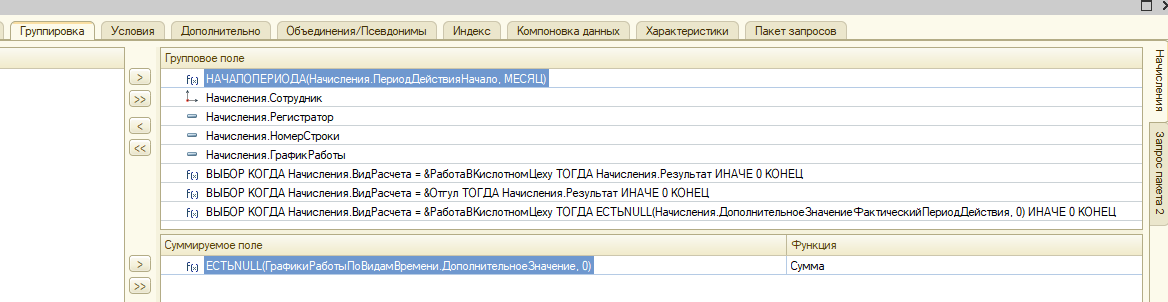
**ПЕРВЫЙ ПАКЕТ**

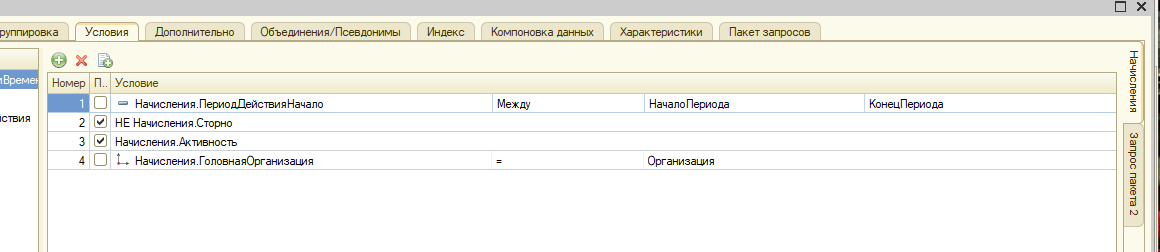
**(ФактическийПериодДействия – из регистра Начисления):**

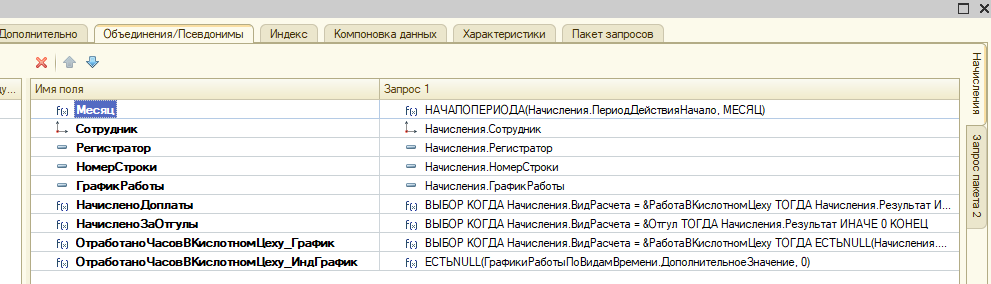


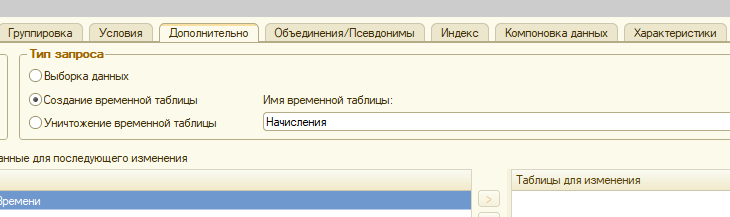








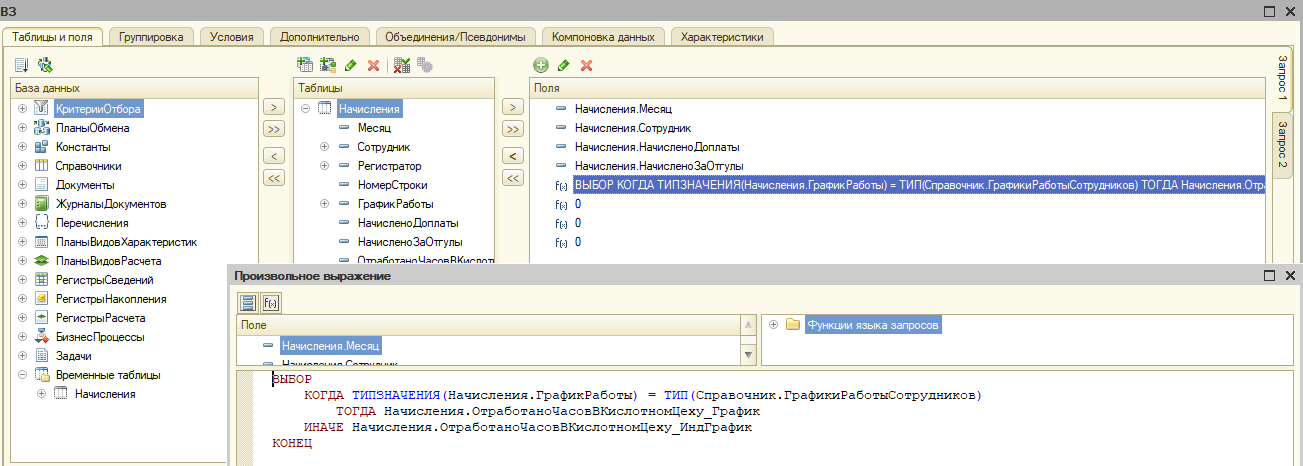




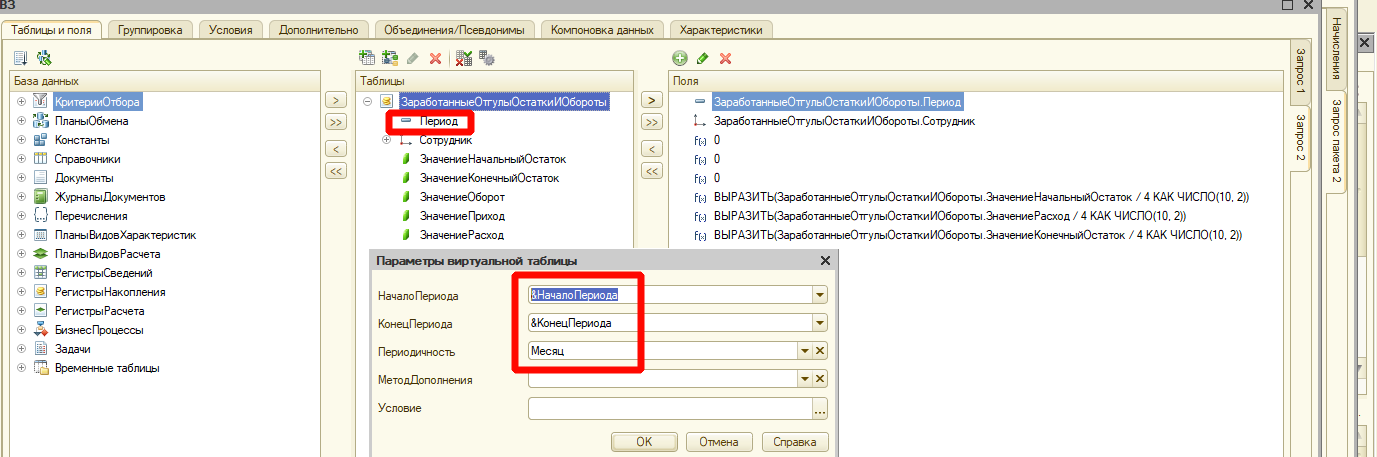
**ВТОРОЙ ПАКЕТ:**

**НАЧАЛО ВЛОЖЕННОГО ЗАПРОСА «ВЗ»:**

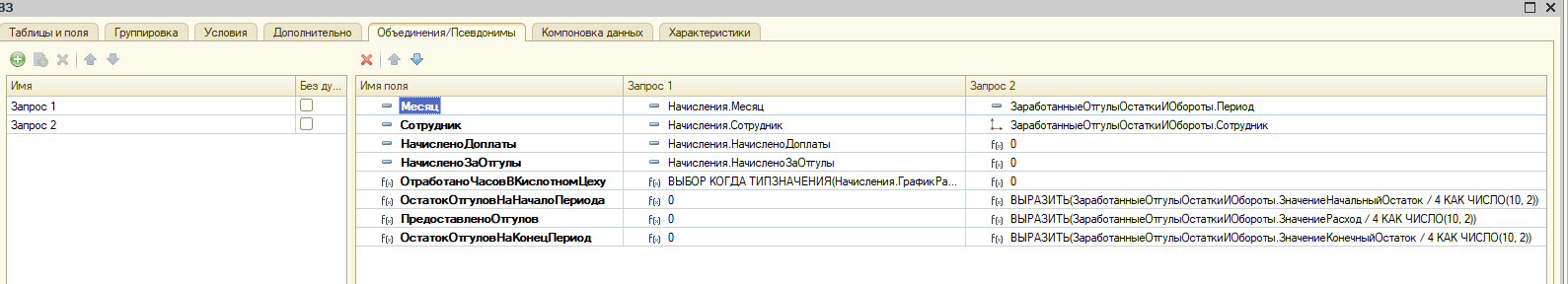
**ПЕРВЫЙ ПАКЕТ:**



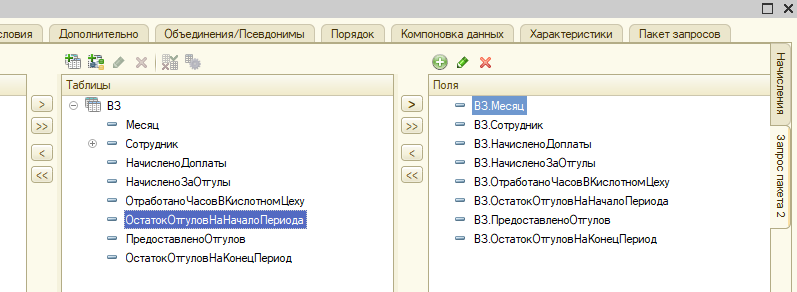
**ВТОРОЙ ПАКЕТ:**

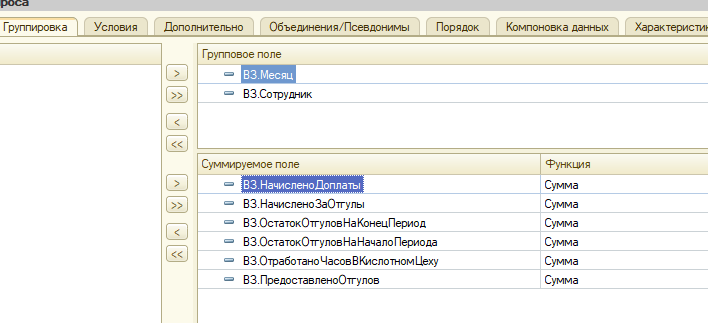


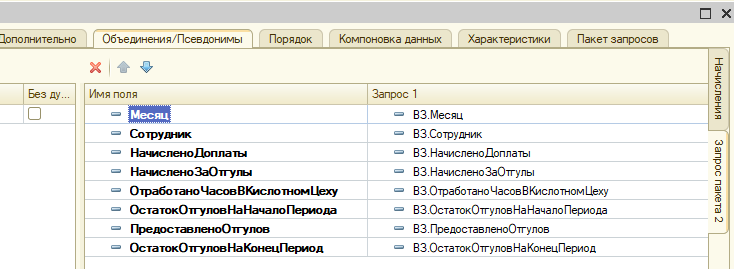
**ОБЪЕДИНЕНИЕ:**



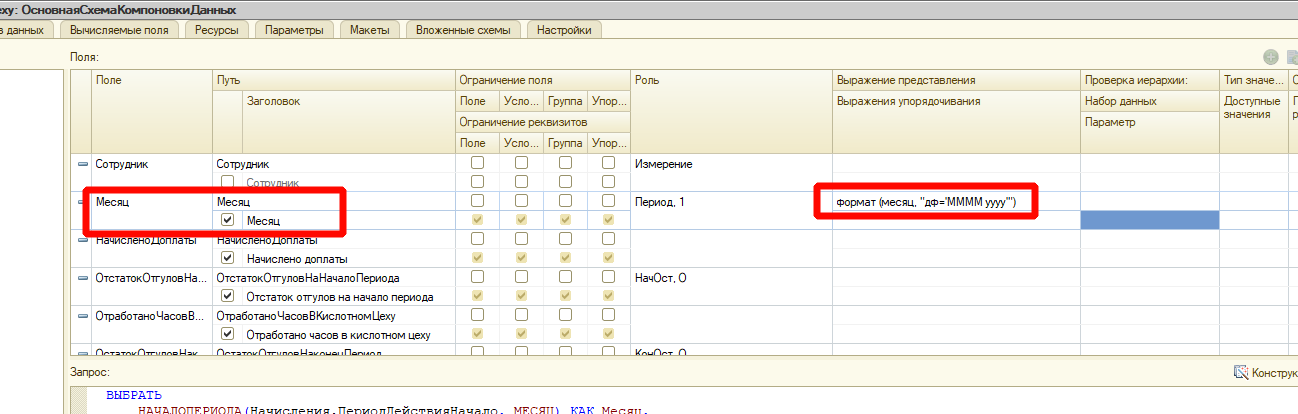
**КОНЕЦ ВЛОЖЕННОГО ЗАПРОСА «ВЗ»**

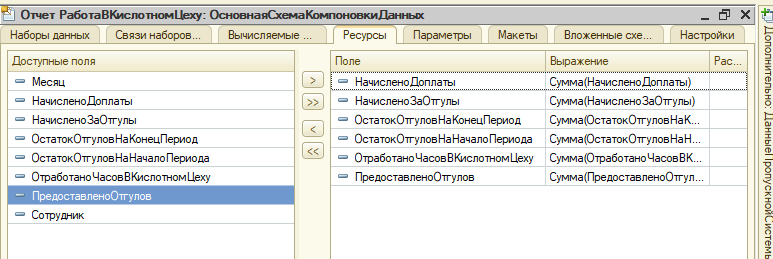


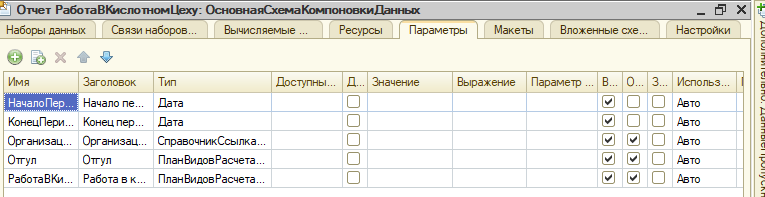


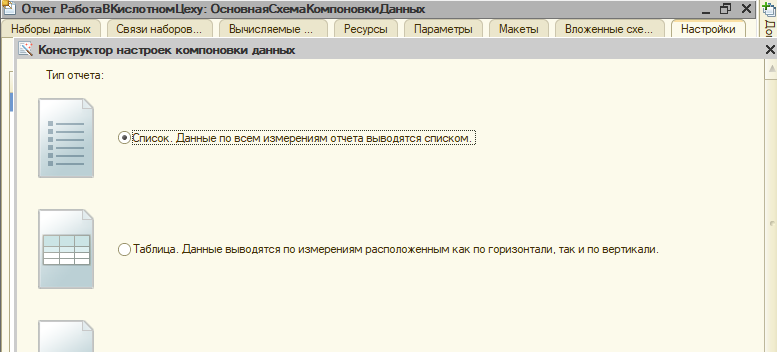


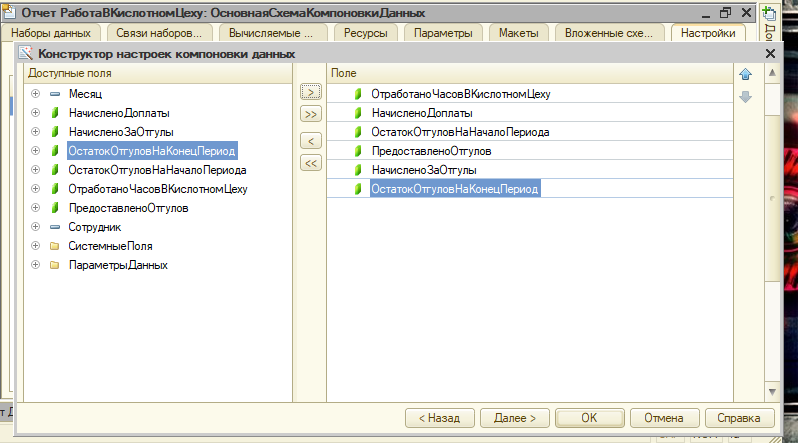
**СКД:**

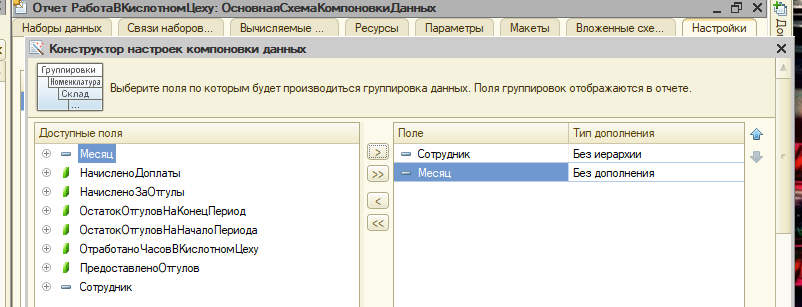




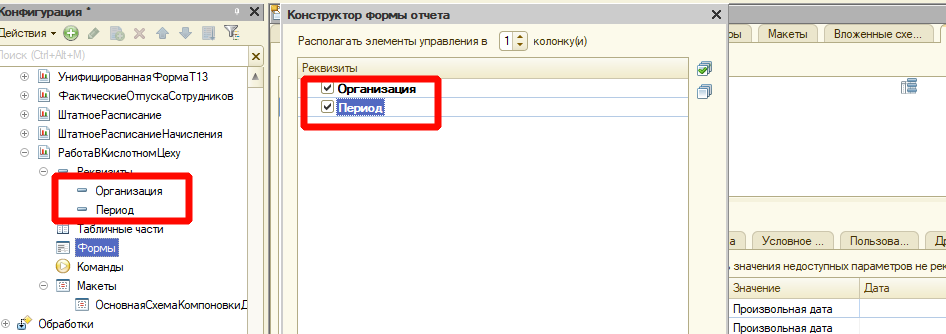


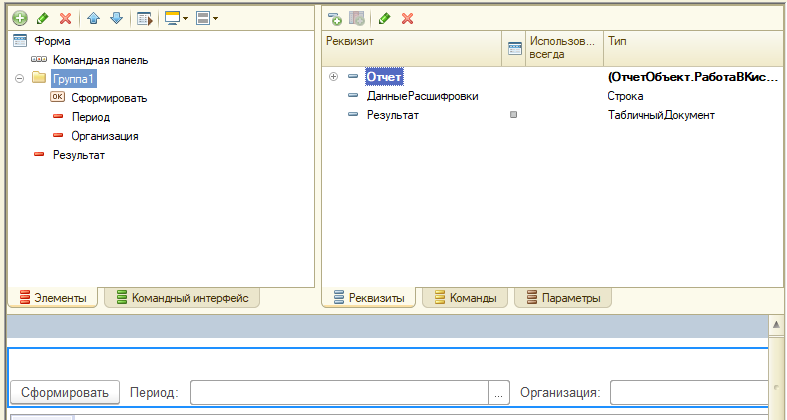




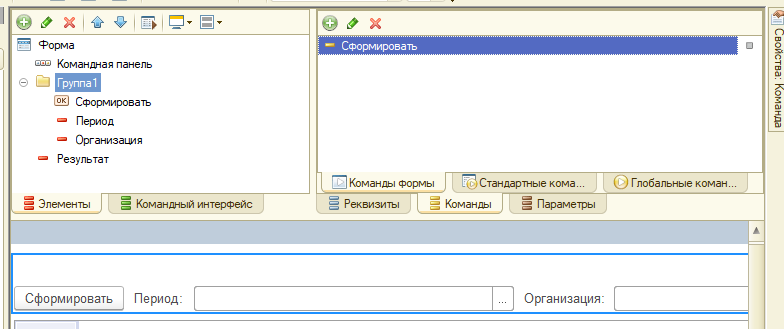


**Добавляем реквизиты (стандартный период и организация) в отчет. Создаем форму:**





**На клиенте и на сервере**



**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("Отгул") Тогда

ПараметрДанных.Использование = Истина;

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

КонецЕсли;

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

ПараметрДанных.Использование = Истина;

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

КонецЕсли;

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

ПараметрДанных.Использование = Истина;

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

КонецЕсли;

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

ПараметрДанных.Использование = Истина;

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

КонецЕсли;

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

ПараметрДанных.Использование = Истина;

ПараметрДанных.Значение = Отчет.Организация;

КонецЕсли;

КонецЦикла;

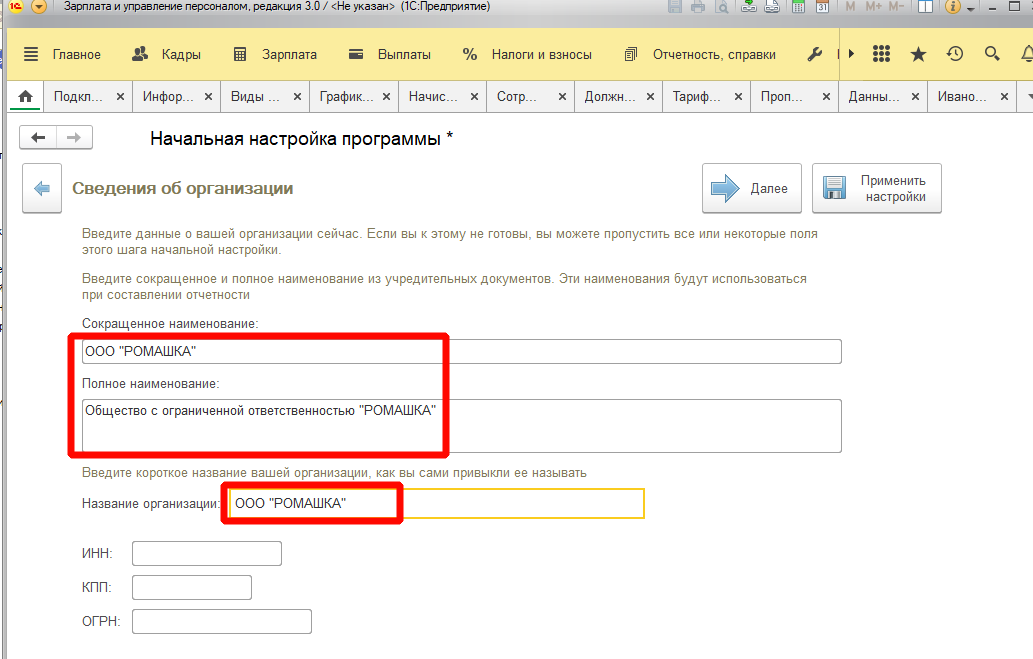
Отчет.КомпоновщикНастроек.ЗагрузитьНастройки(Настройки);

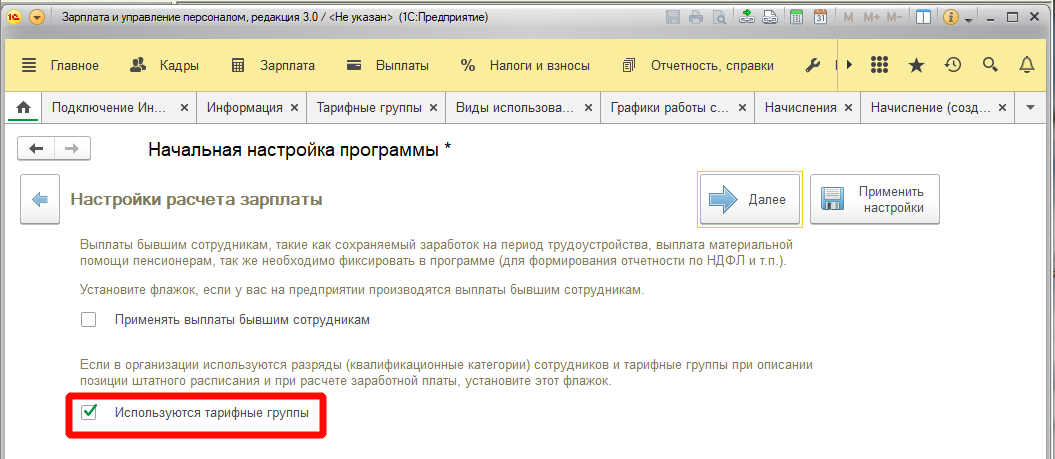
СкомпоноватьРезультат();

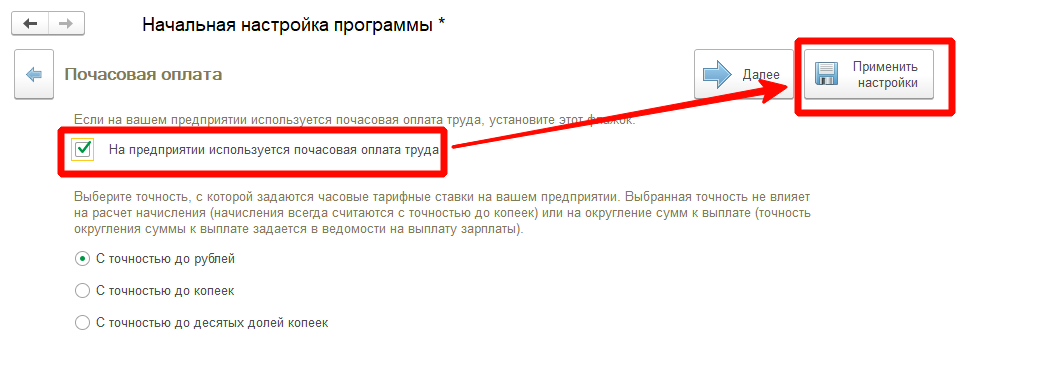
КонецПроцедуры

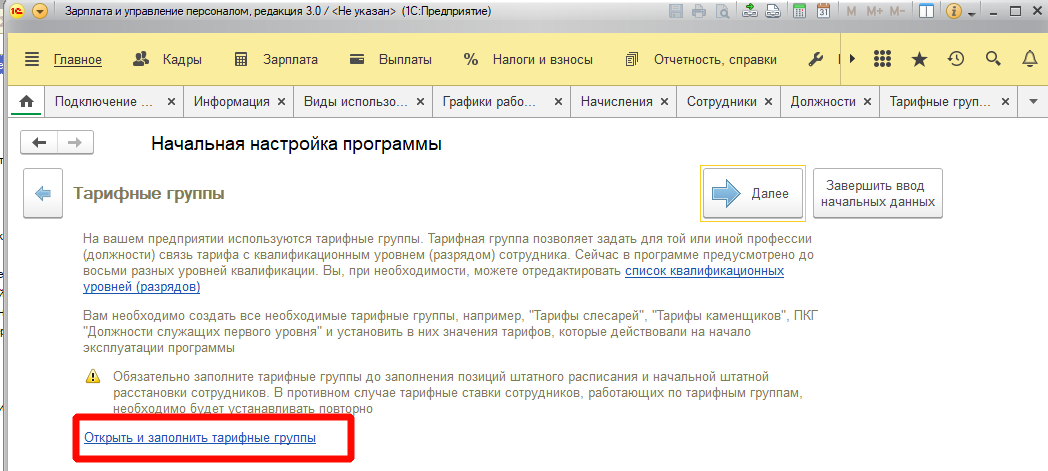
**III Ввод контрольного примера и проверка решения:**

1. **На начальной странице или все функции-обработки-начальная настройка программы:**

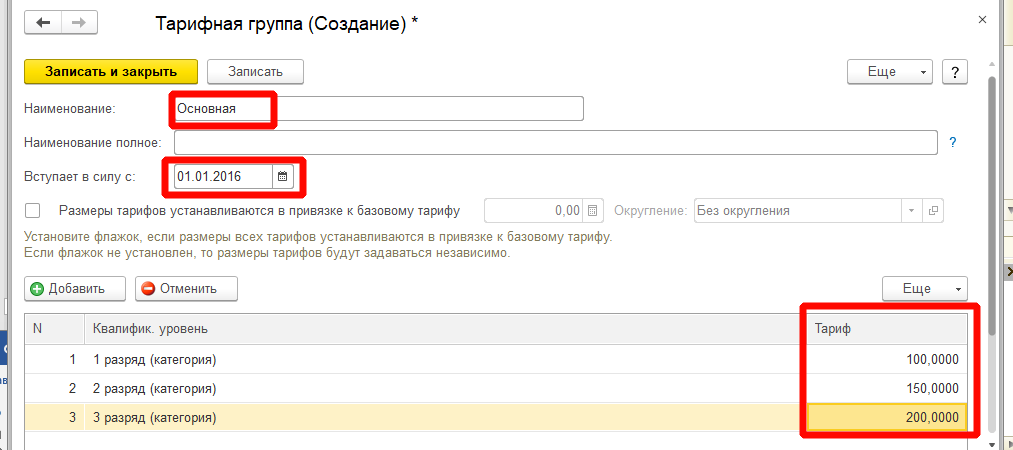


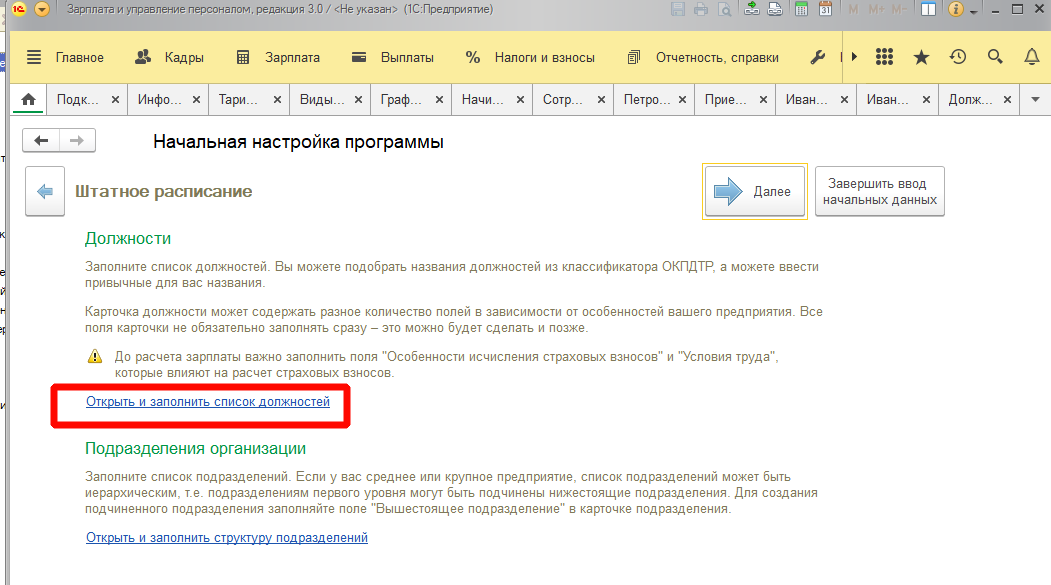




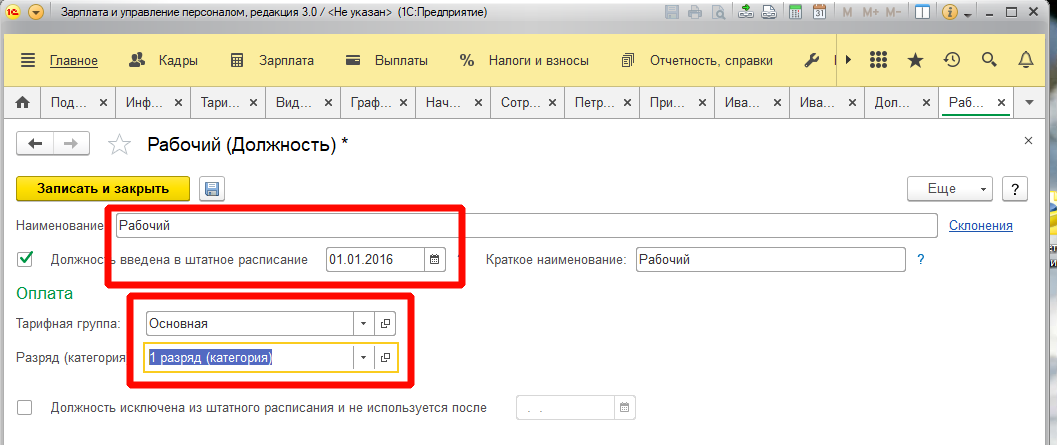


**Создаем тарифную группу:**

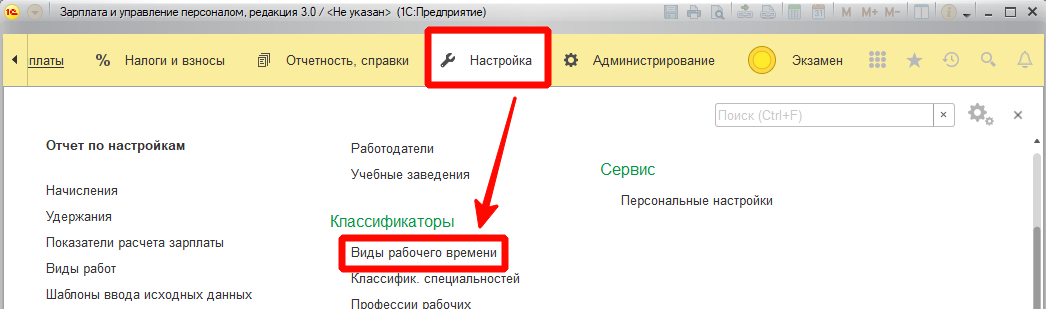


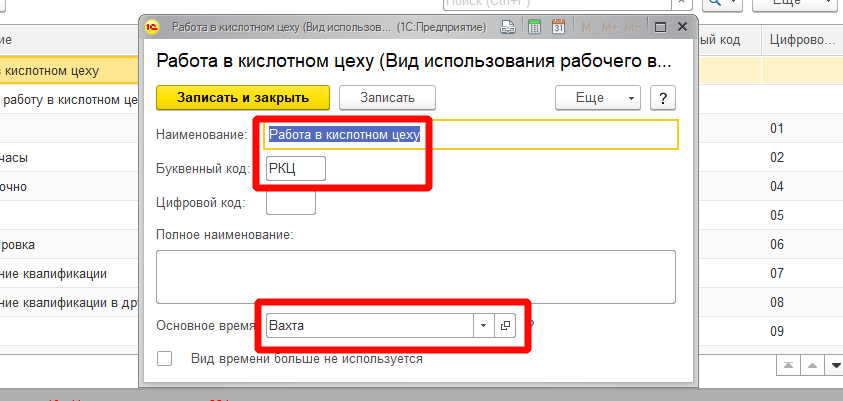


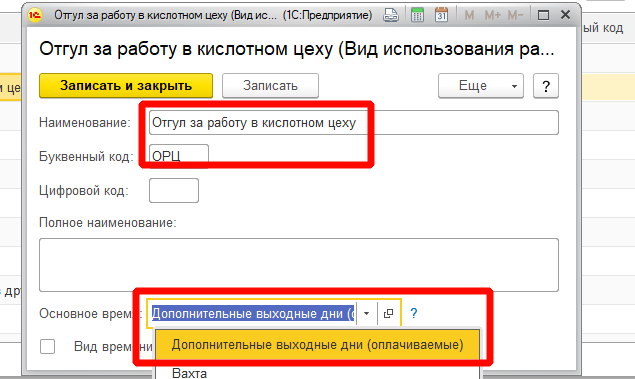
**Создаем рабочего**



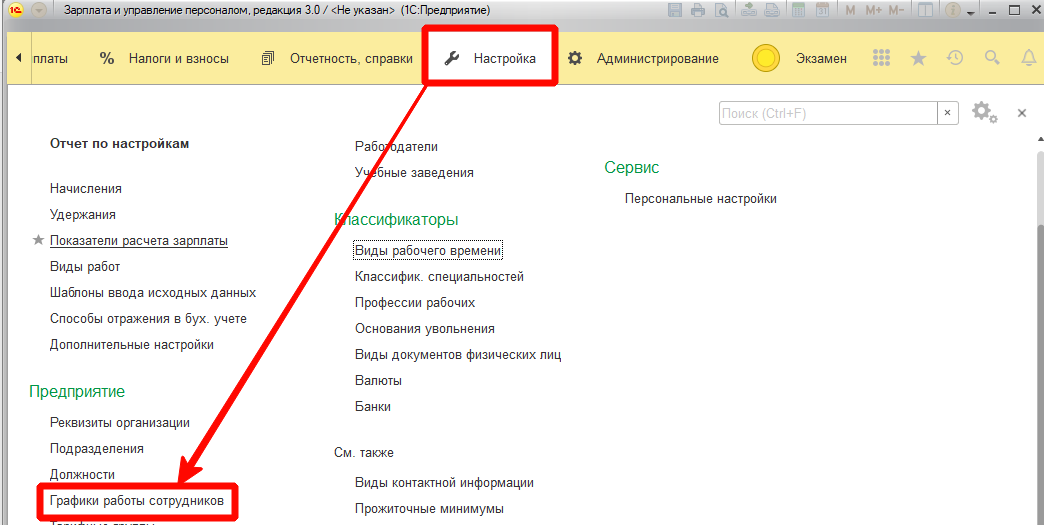
1. **Добавляем два Вида использования рабочего времени:**

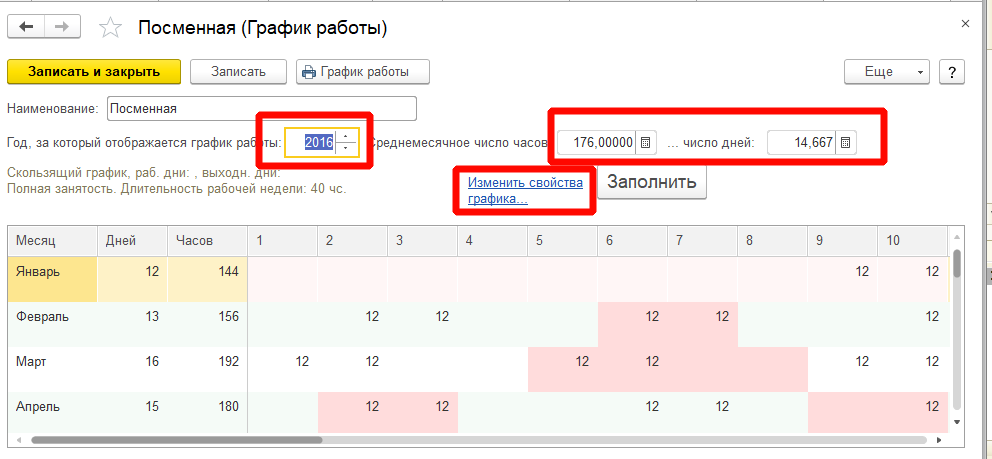


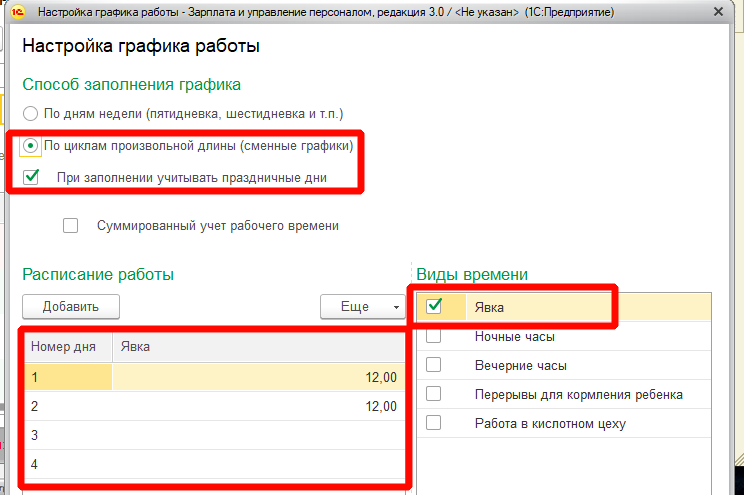




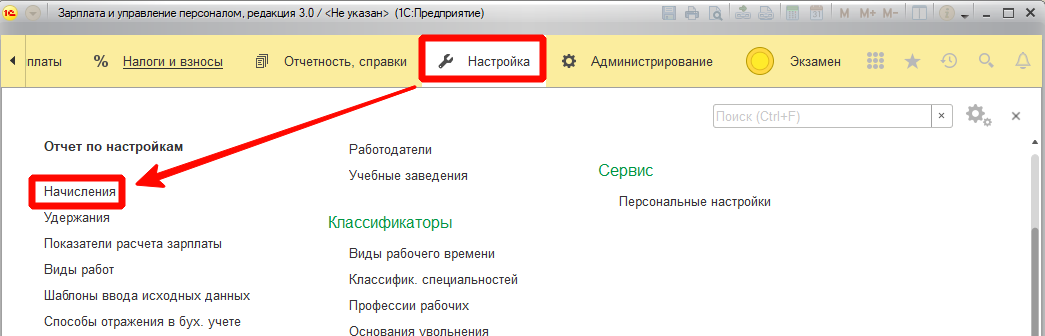
1. **Добавляем «Посменный» график:**

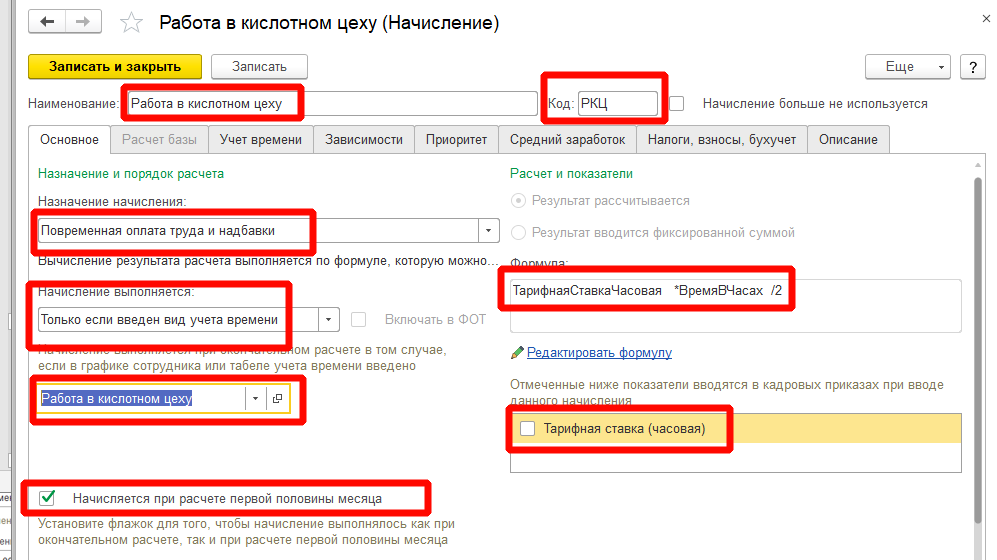


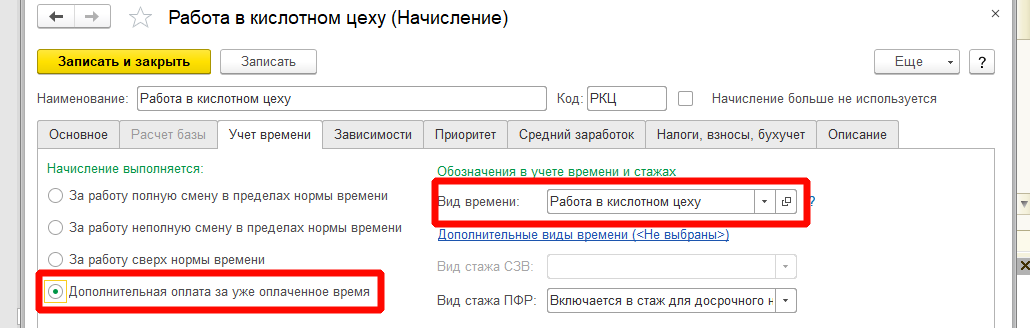


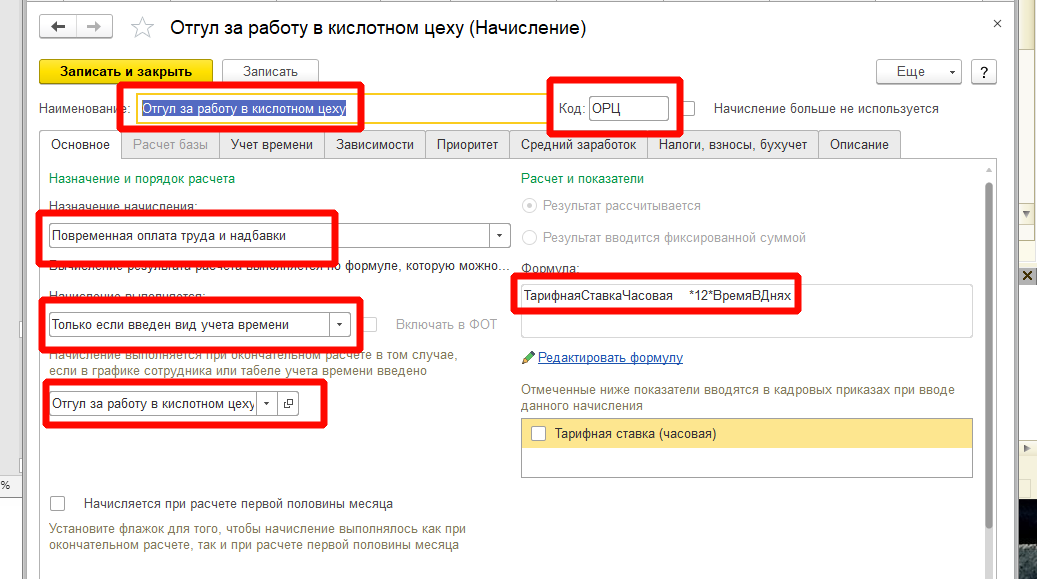


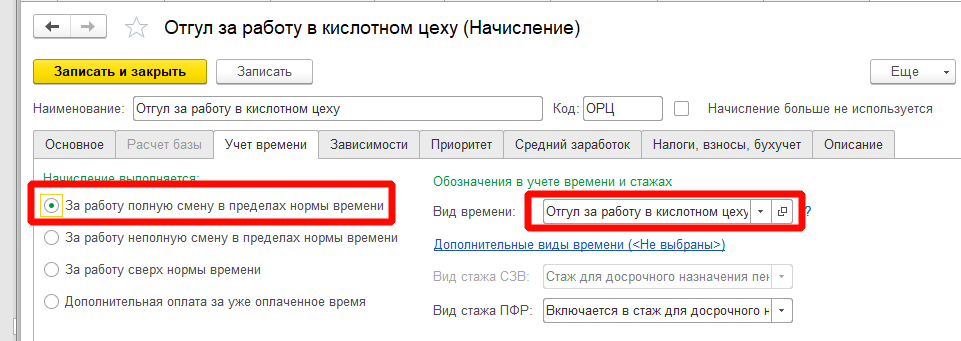
1. **Добавляем начисления «Работа в кислотном цеху» и «Отгул за работу в кислотном цеху»:**



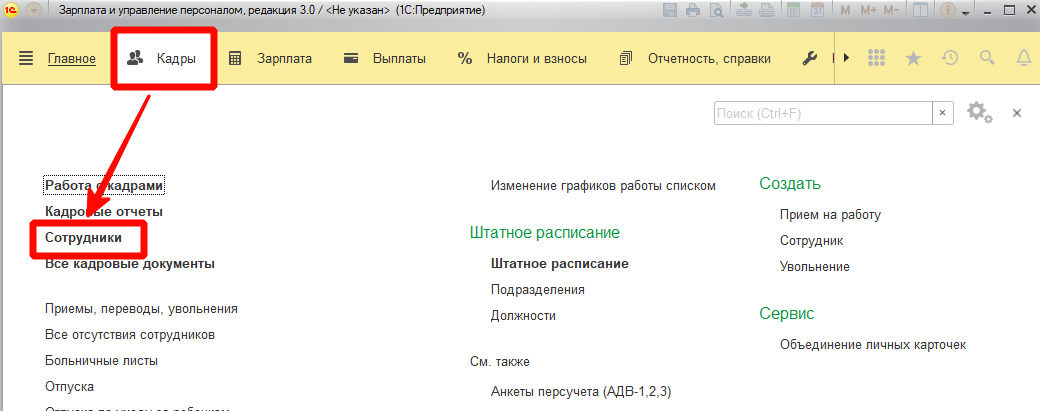


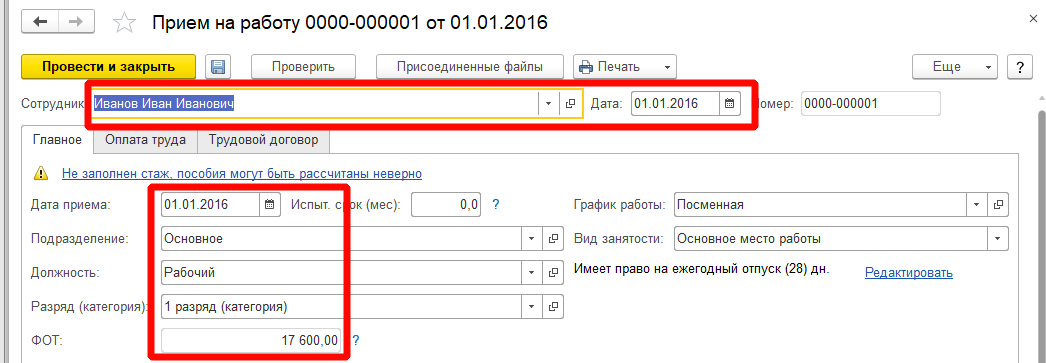


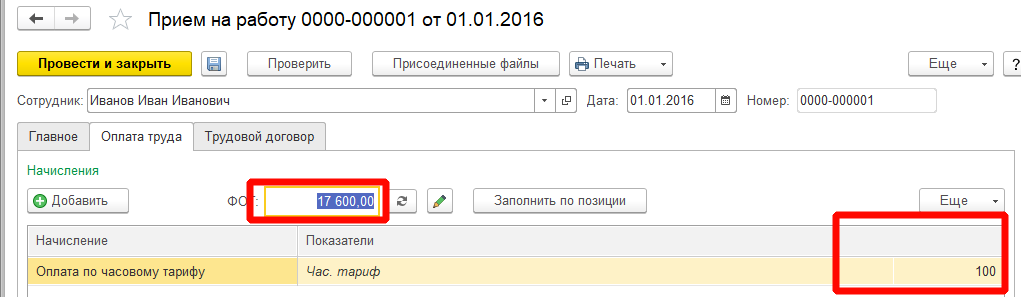


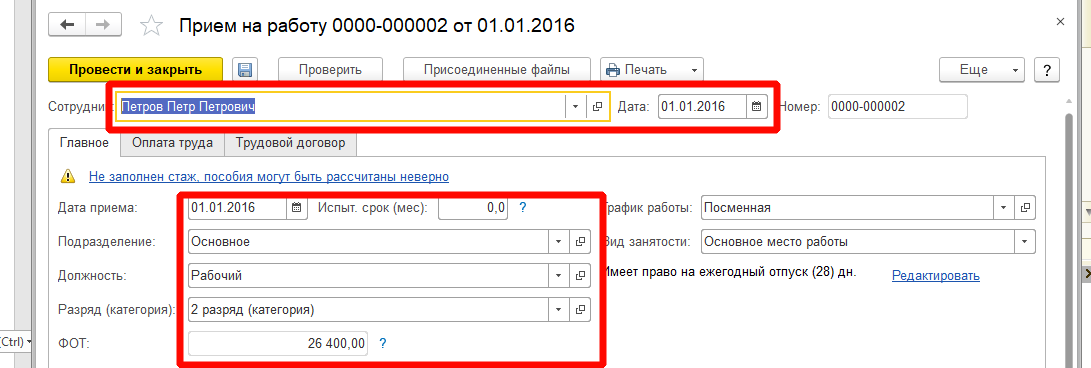


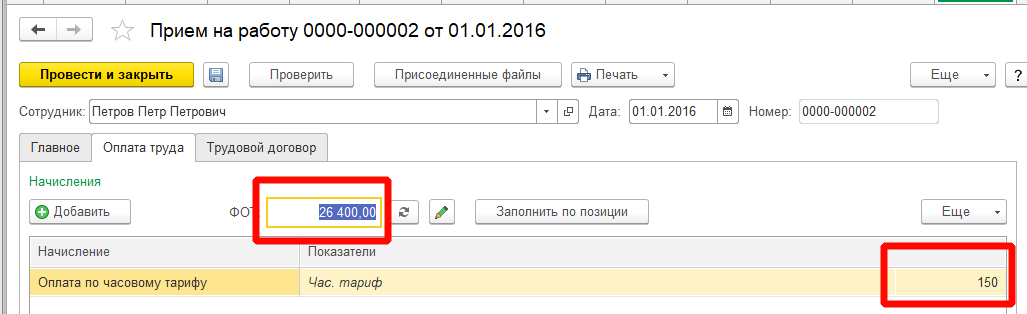
1. **Создаем двух сотрудников - Иванова Ивана Ивановича и Петрова Петра Петровича и принимаем на работу:**



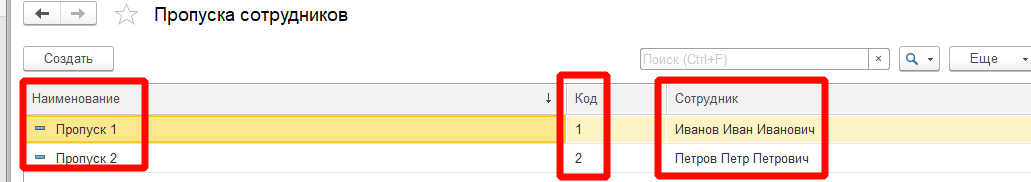




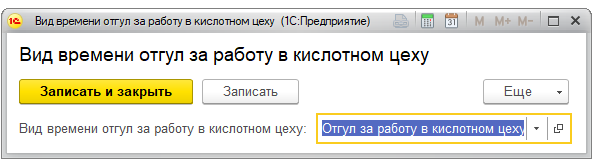


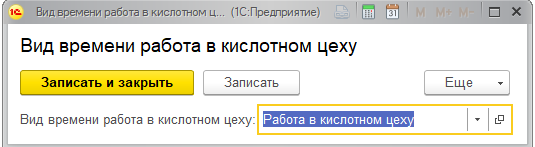


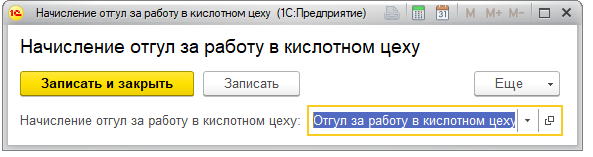
1. **Экзамен-пропуска сотрудников - вводим пропуска сотрудников:**

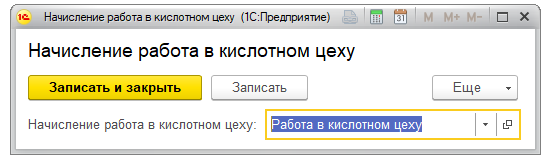


1. **Экзамен-сервис- заполняем четыре константы:**

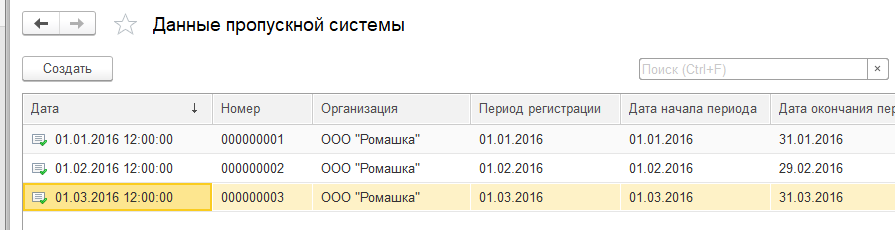


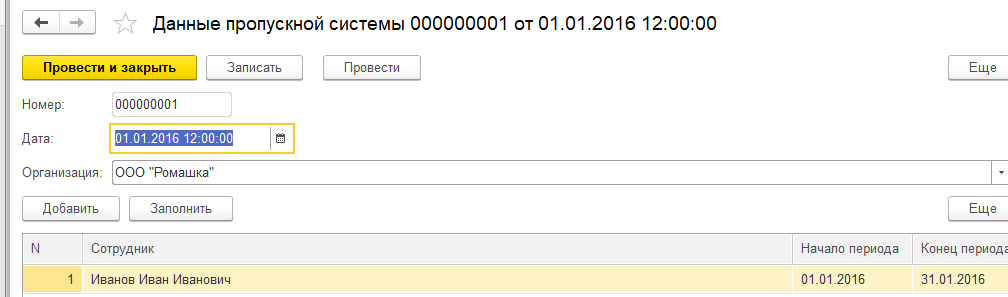


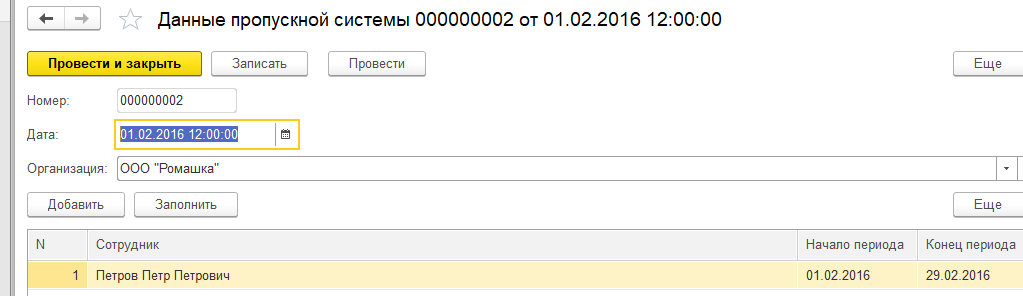


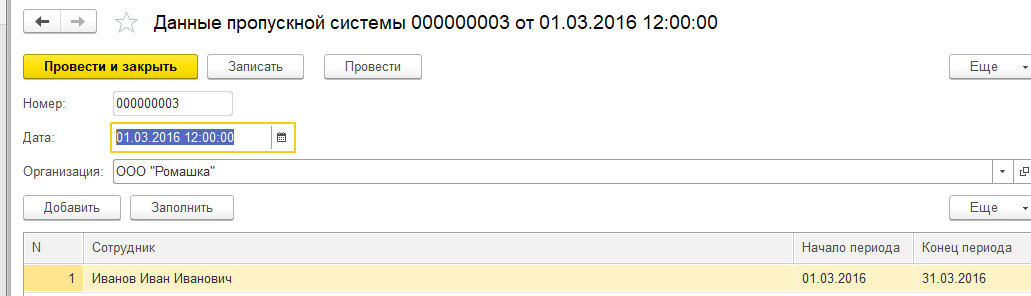


1. **Экзамен – документы – данные пропускной системы:**

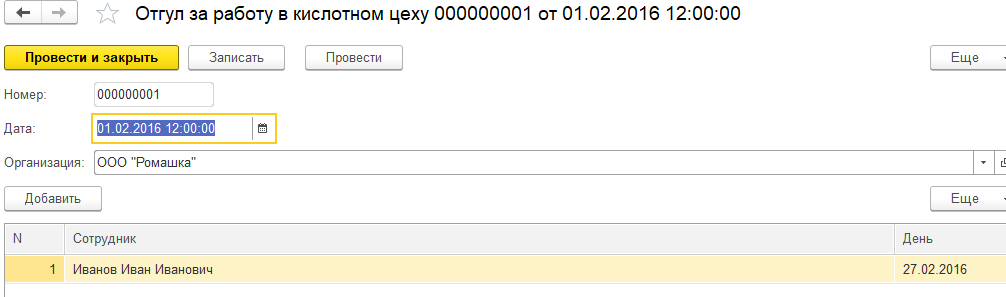


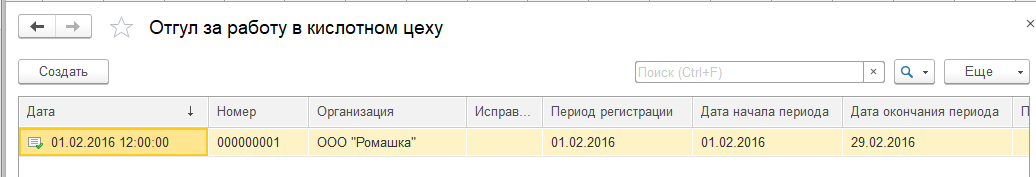




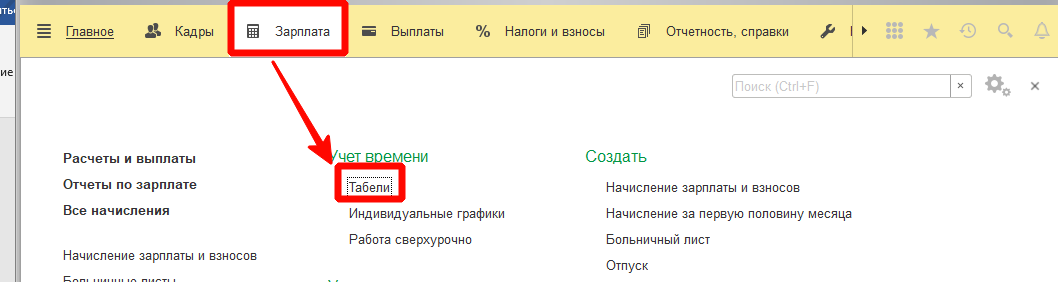


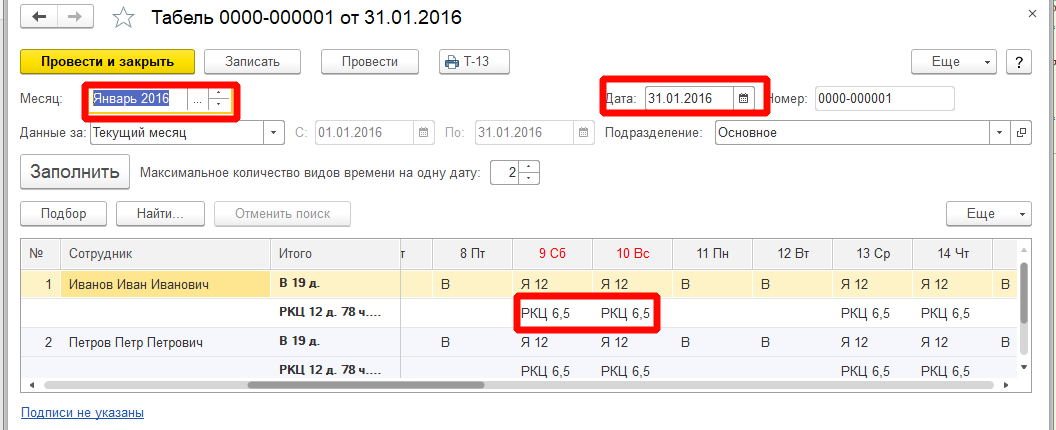
1. **Экзамен-документы-отгул за работу в кислотном цеху:**

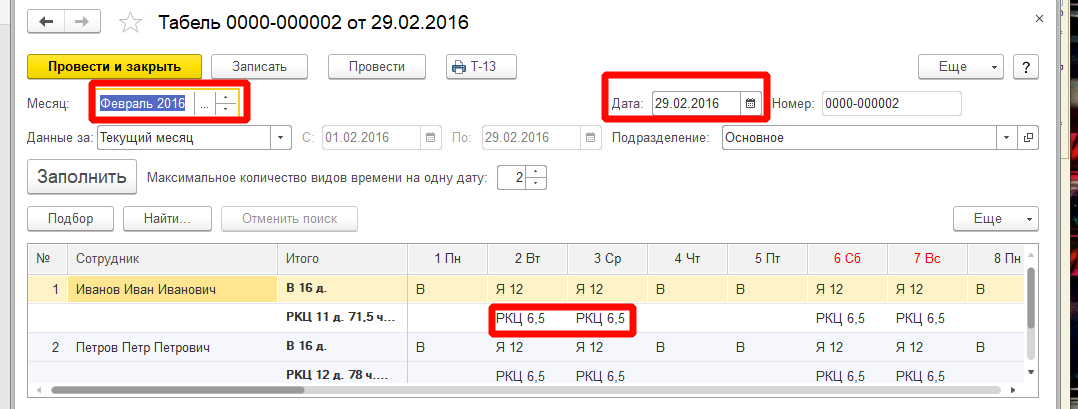


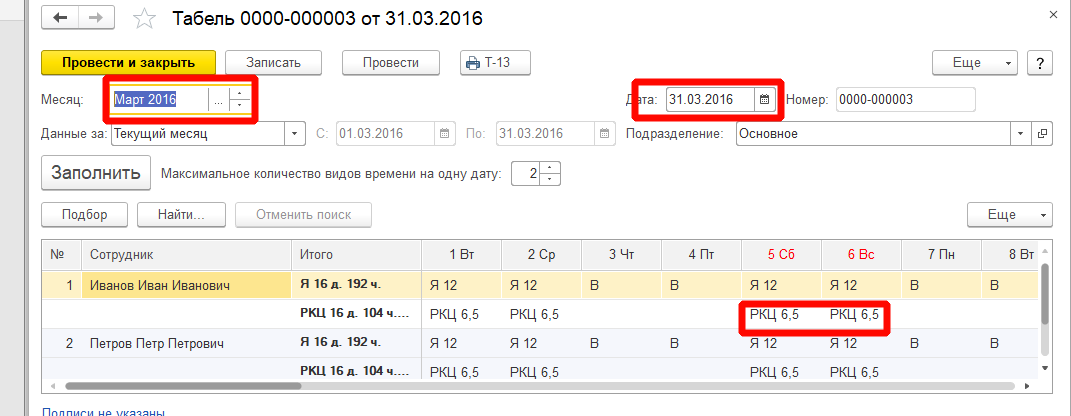


1. **Вводим табели за январь, февраль и март 2016г. Везде где явка- ставить РКЦ 6,5:**



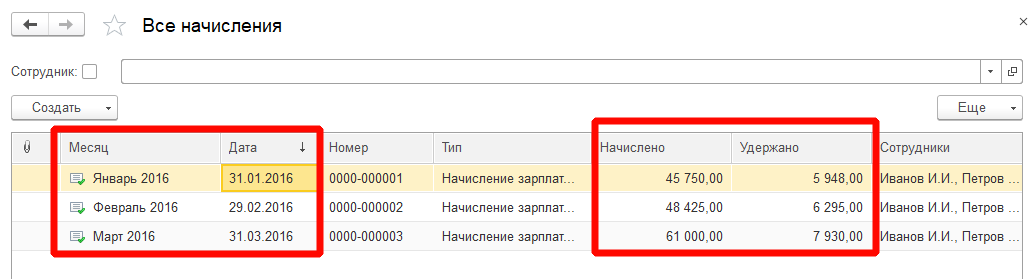




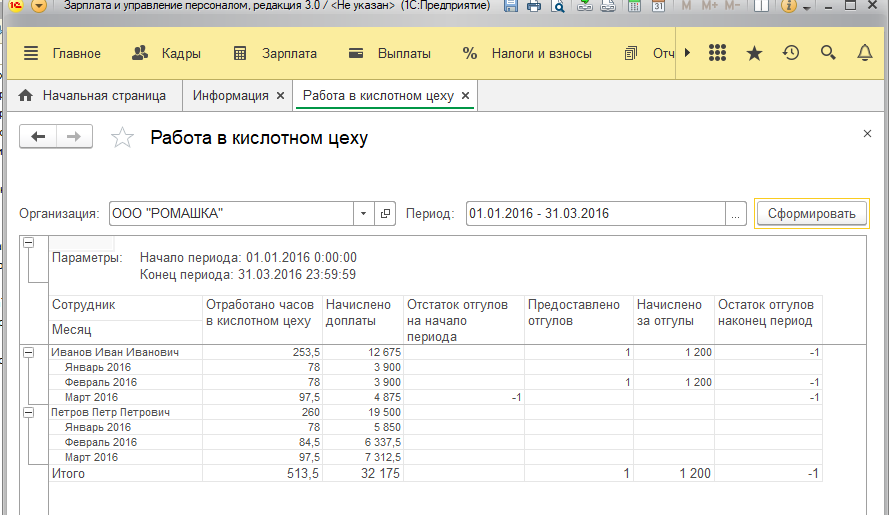


1. **Начисляем зп за январь, февраль и март 2016г.:**

**Если цифры совпадают, то все ок.**



1. **В завершении смотрим отчет:**



**IV ПОЯСНЕНИЕ К РЕШЕНИЮ**

Для реализации решения созданы константы для видов времени и начислений.

Создан регистр накопления ЗаработанныеОтгулы для хранения количества отработанных часов в кислотном цеху. При списании из регистра остатки не контролирую т.к. такого условия в задаче нет.

Создан документ ДанныеПропускнойСистемы с загрузкой данных из текстового файла в соответствии с заданием. Документ формирует движения в регистры ДанныеОперативногоУчетаРабочегоВремениСотрудников для регистрации отработанного времени и ЗаработанныеОтгулы для регистрации отгулов.

Создан документ ОтгулЗаРаботуВКислотномЦеху, который формирует движения в регистры накопления ДанныеОперативногоУчетаРабочегоВремениСотрудников для регистрации отгулов и ЗаработанныеОтгулы с расходом по заработанным отгулам.

Создан отчет РаботаВКислотномЦеху в соответствии с заданием. 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.

**V ТЕКСТОВЫЕ ФАЙЛЫ**

|  |  |  |
| --- | --- | --- |
| Работа в кислотном цеху за Январь 2016 | Работа в кислотном цеху за Февраль 2016 | Работа в кислотном цеху за Март 2016 |
| 1;09.01.2016 10:00;09.01.2016 16:30  1;10.01.2016 10:00;10.01.2016 16:30  1;13.01.2016 10:00;13.01.2016 16:30  1;14.01.2016 10:00;14.01.2016 16:30  1;17.01.2016 10:00;17.01.2016 16:30  1;18.01.2016 10:00;18.01.2016 16:30  1;21.01.2016 10:00;21.01.2016 16:30  1;22.01.2016 10:00;22.01.2016 16:30  1;25.01.2016 10:00;25.01.2016 16:30  1;26.01.2016 10:00;26.01.2016 16:30  1;29.01.2016 10:00;29.01.2016 16:30  1;30.01.2016 10:00;30.01.2016 16:30  2;09.01.2016 10:00;09.01.2016 16:30  2;10.01.2016 10:00;10.01.2016 16:30  2;13.01.2016 10:00;13.01.2016 16:30  2;14.01.2016 10:00;14.01.2016 16:30  2;17.01.2016 10:00;17.01.2016 16:30  2;18.01.2016 10:00;18.01.2016 16:30  2;21.01.2016 10:00;21.01.2016 16:30  2;22.01.2016 10:00;22.01.2016 16:30  2;25.01.2016 10:00;25.01.2016 16:30  2;26.01.2016 10:00;26.01.2016 16:30  2;29.01.2016 10:00;29.01.2016 16:30  2;30.01.2016 10:00;30.01.2016 16:30 | 1;02.02.2016 10:00;02.02.2016 16:30  1;03.02.2016 10:00;03.02.2016 16:30  1;06.02.2016 10:00;06.02.2016 16:30  1;07.02.2016 10:00;07.02.2016 16:30  1;10.02.2016 10:00;10.02.2016 16:30  1;11.02.2016 10:00;11.02.2016 16:30  1;14.02.2016 10:00;14.02.2016 16:30  1;15.02.2016 10:00;15.02.2016 16:30  1;18.02.2016 10:00;18.02.2016 16:30  1;19.02.2016 10:00;19.02.2016 16:30  1;21.02.2016 10:00;21.02.2016 16:30  1;22.02.2016 10:00;22.02.2016 16:30  1;26.02.2016 10:00;26.02.2016 16:30  1;27.02.2016 10:00;27.02.2016 16:30  2;02.02.2016 10:00;02.02.2016 16:30  2;03.02.2016 10:00;03.02.2016 16:30  2;06.02.2016 10:00;06.02.2016 16:30  2;07.02.2016 10:00;07.02.2016 16:30  2;10.02.2016 10:00;10.02.2016 16:30  2;11.02.2016 10:00;11.02.2016 16:30  2;14.02.2016 10:00;14.02.2016 16:30  2;15.02.2016 10:00;15.02.2016 16:30  2;18.02.2016 10:00;18.02.2016 16:30  2;19.02.2016 10:00;19.02.2016 16:30  2;21.02.2016 10:00;21.02.2016 16:30  2;22.02.2016 10:00;22.02.2016 16:30  2;26.02.2016 10:00;26.02.2016 16:30  2;27.02.2016 10:00;27.02.2016 16:30 | 1;01.03.2016 10:00;01.03.2016 16:30  1;02.03.2016 10:00;02.03.2016 16:30  1;05.03.2016 10:00;05.03.2016 16:30  1;06.03.2016 10:00;06.03.2016 16:30  2;01.03.2016 10:00;01.03.2016 16:30  2;02.03.2016 10:00;02.03.2016 16:30  2;05.03.2016 10:00;05.03.2016 16:30  2;06.03.2016 10:00;06.03.2016 16:30 |