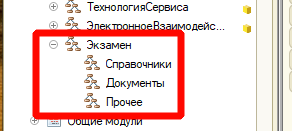
**БИЛЕТ 8 (или Билет 2 из новых билетов)**

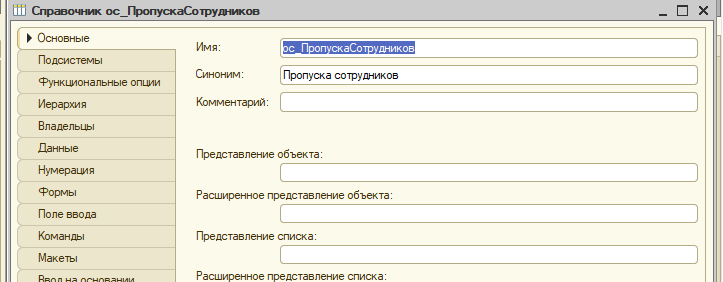
**РЕШЕНИЕ:**

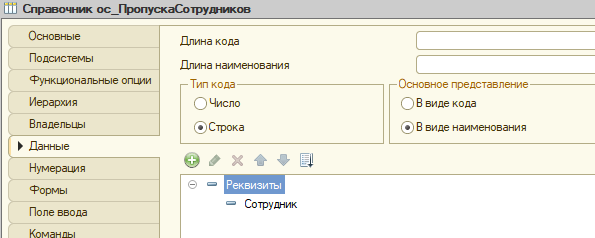
**I Конфигуратор.**

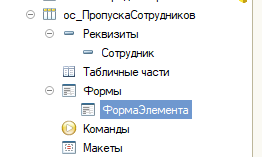
1. **Снимаем с поддержки, создаем подсистемы:**



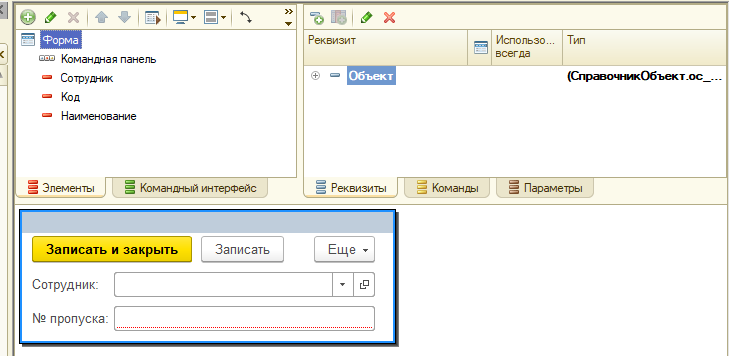
1. **Создаем справочник ос\_ПропускаСотрудников:**

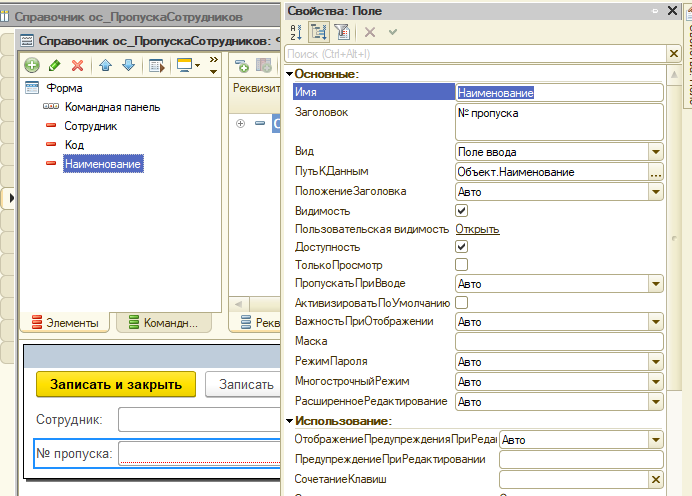




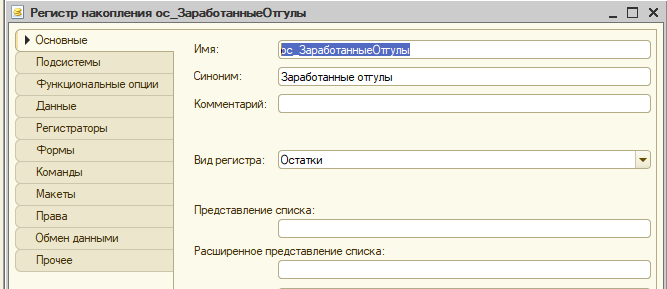


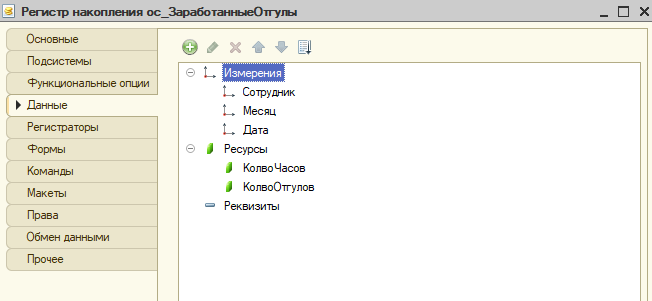
**Форма обычная:**





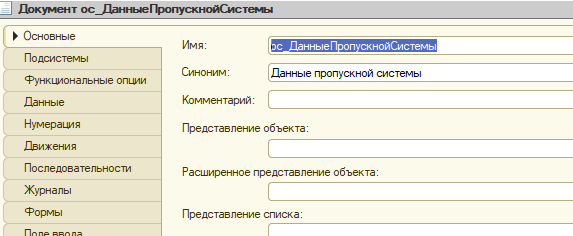
1. **Создаем регистр накопления ос\_ЗаработанныеОтгулы:**

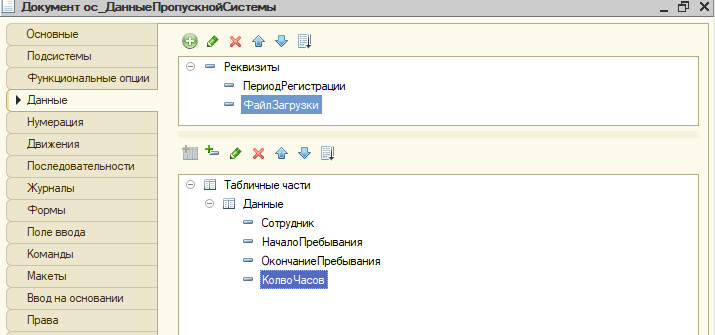


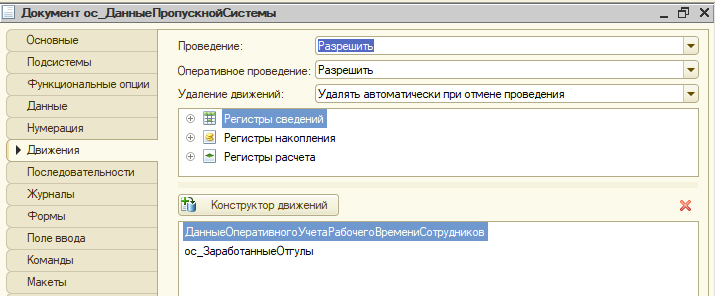


**Месяц, дата – тип «дата», ресурсы – «числа»**

1. **Создаем документ ос\_ДанныеПропускнойСистемы:**

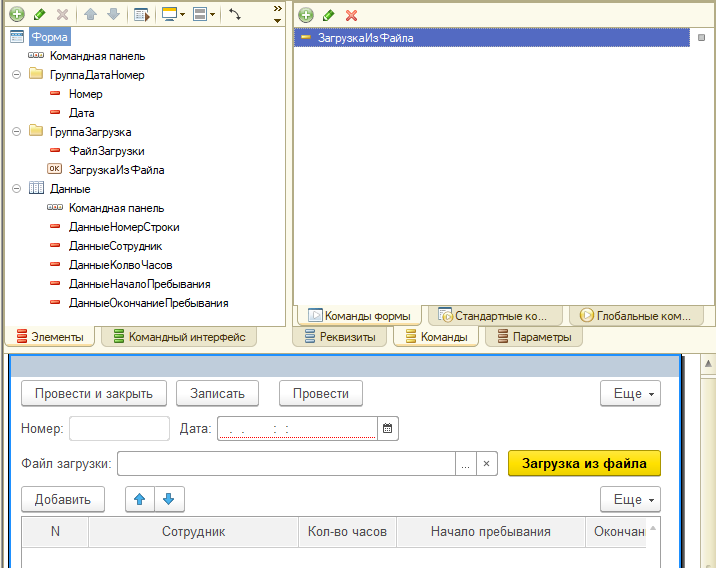




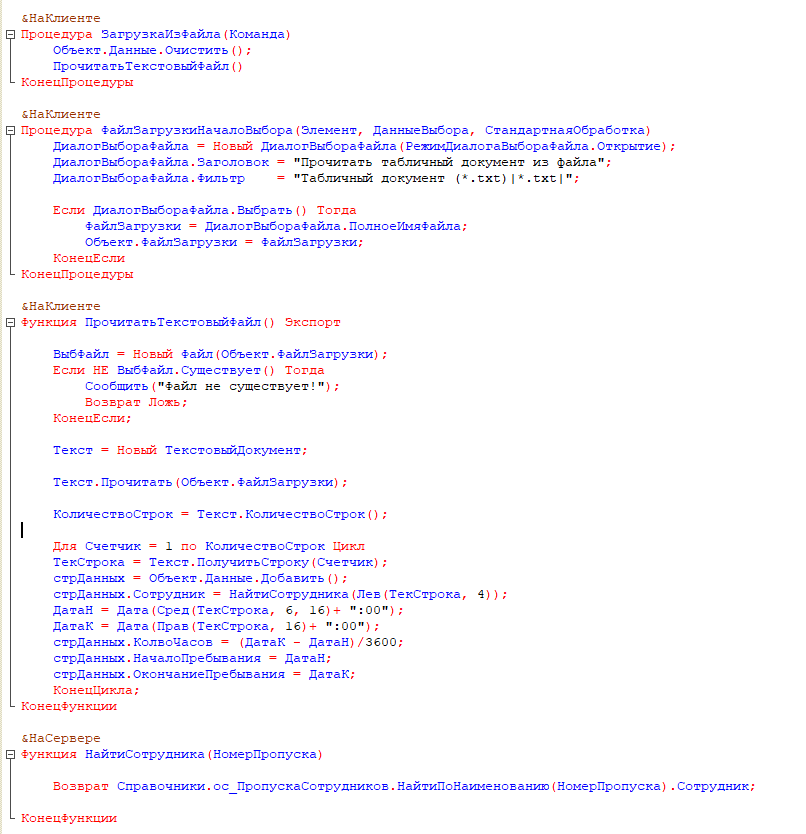


**Создаем форму и команду ЗагрузкаИзФайла (на клиенте)**

**и для начала выбора «ФайлЗагрузка» событие на клиенте:**



**Модуль формы ос\_ДанныеПропускнойСистемы:**



&НаКлиенте

Процедура ЗагрузкаИзФайла(Команда)

Объект.Данные.Очистить();

ПрочитатьТекстовыйФайл()

КонецПроцедуры

&НаКлиенте

Процедура ФайлЗагрузкиНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

ДиалогВыбораФайла = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

ДиалогВыбораФайла.Заголовок = "Прочитать табличный документ из файла";

ДиалогВыбораФайла.Фильтр = "Табличный документ (\*.txt)|\*.txt|";

Если ДиалогВыбораФайла.Выбрать() Тогда

ФайлЗагрузки = ДиалогВыбораФайла.ПолноеИмяФайла;

Объект.ФайлЗагрузки = ФайлЗагрузки;

КонецЕсли

КонецПроцедуры

&НаКлиенте

Функция ПрочитатьТекстовыйФайл() Экспорт

ВыбФайл = Новый Файл(Объект.ФайлЗагрузки);

Если НЕ ВыбФайл.Существует() Тогда

Сообщить("Файл не существует!");

Возврат Ложь;

КонецЕсли;

Текст = Новый ТекстовыйДокумент;

Текст.Прочитать(Объект.ФайлЗагрузки);

КоличествоСтрок = Текст.КоличествоСтрок();

Для Счетчик = 1 по КоличествоСтрок Цикл

ТекСтрока = Текст.ПолучитьСтроку(Счетчик);

стрДанных = Объект.Данные.Добавить();

стрДанных.Сотрудник = НайтиСотрудника(Лев(ТекСтрока, 4));

ДатаН = Дата(Сред(ТекСтрока, 6, 16)+ ":00");

ДатаК = Дата(Прав(ТекСтрока, 16)+ ":00");

стрДанных.КолвоЧасов = (ДатаК - ДатаН)/3600;

стрДанных.НачалоПребывания = ДатаН;

стрДанных.ОкончаниеПребывания = ДатаК;

КонецЦикла;

КонецФункции

&НаСервере

Функция НайтиСотрудника(НомерПропуска)

Возврат Справочники.ос\_ПропускаСотрудников.НайтиПоНаименованию(НомерПропуска).Сотрудник;

КонецФункции

**Модуль документа ос\_ДанныеПропускнойСистемы:**



Процедура ОбработкаПроведения(Отказ, Режим)

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ос\_ЗаработанныеОтгулы.Записывать = Истина;

Для Каждого ТекСтрокаДанные Из Данные Цикл

// регистр ДанныеОперативногоУчетаРабочегоВремениСотрудников

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = НачалоДня(ТекСтрокаДанные.НачалоПребывания);

Движение.Сотрудник = ТекСтрокаДанные.Сотрудник;

Движение.ПериодРегистрации = НачалоМесяца(ТекСтрокаДанные.НачалоПребывания);

Движение.ВидУчетаВремени = Справочники.ВидыИспользованияРабочегоВремени.НайтиПоНаименованию("Работа на складе-холодильнике");

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДополнительноеВнутрисменноеВремя;

Движение.Дни = 1;

Движение.Часы = ТекСтрокаДанные.КолвоЧасов;

Движение.Организация = Справочники.Организации.НайтиПоКоду(1);

// регистр ос\_ЗаработанныеОтгулы Приход

Движение1 = Движения.ос\_ЗаработанныеОтгулы.Добавить();

Движение1.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение1.Период = НачалоДня(ТекСтрокаДанные.НачалоПребывания);

Движение1.Сотрудник = ТекСтрокаДанные.Сотрудник;

Движение1.Месяц = НачалоМесяца(ТекСтрокаДанные.НачалоПребывания);

Движение1.Дата = НачалоДня(ТекСтрокаДанные.НачалоПребывания);

Движение1.КолвоЧасов = ТекСтрокаДанные.КолвоЧасов;

Движение1.КолвоОтгулов = Окр(ТекСтрокаДанные.КолвоЧасов / 8, 1); // ???

КонецЦикла;

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

//присваивает дату выгрузки файла ДанныеПропускнойСистемы

Если Данные.Количество()<> 0 Тогда

//Дата = Дата(Год(Данные[0].Дата), Месяц(Данные[0].Дата), День(КонецМесяца(Данные[0].Дата)));

Если День(Данные[0].НачалоПребывания) < 15 Тогда

Дата = Дата(Год(Данные[0].НачалоПребывания), Месяц(Данные[0].НачалоПребывания), 15);

Иначе

Дата = Дата(Год(Данные[0].НачалоПребывания), Месяц(Данные[0].НачалоПребывания), День(КонецМесяца(Данные[0].НачалоПребывания)));

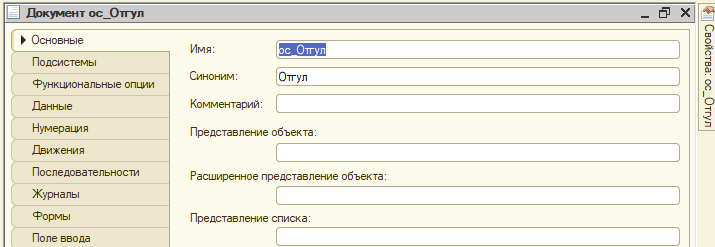
КонецЕсли;

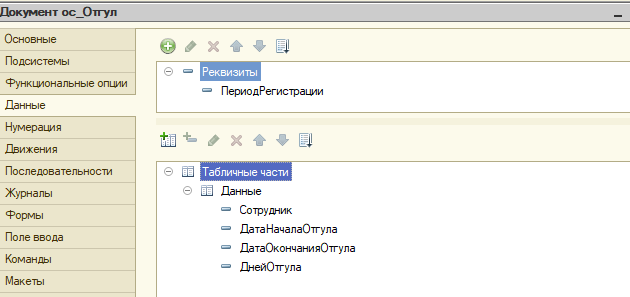
ЭтотОбъект.ПериодРегистрации = НачалоМесяца(Данные[0].НачалоПребывания);

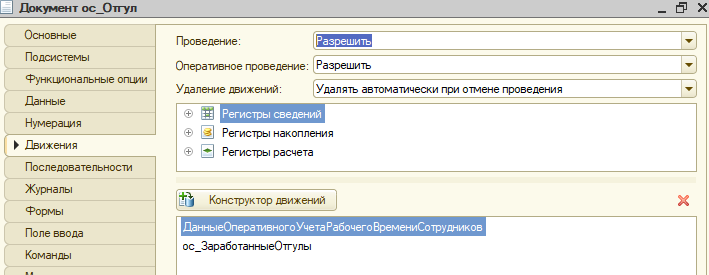
КонецЕсли;

КонецПроцедуры

1. **Создаем Документ ос\_Отгул (экзамен-документы):**

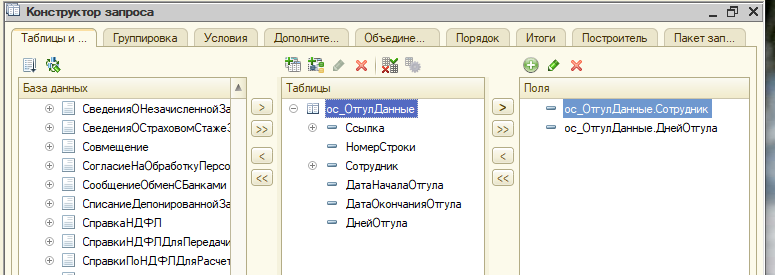


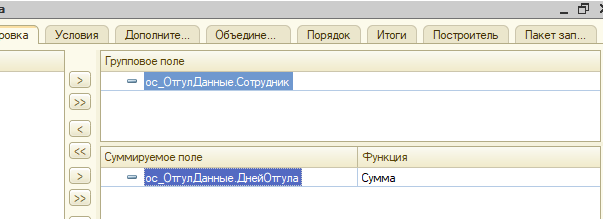


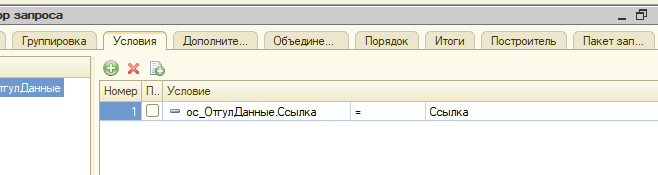


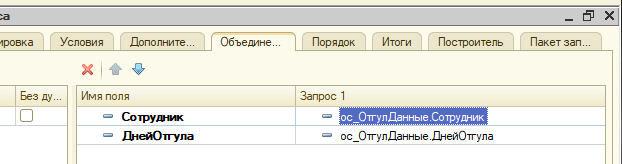
**МОДУЛЬ ДОКУМЕНТА ос\_Отгул:**

**ПЕРВЫЙ ЗАПРОС:**

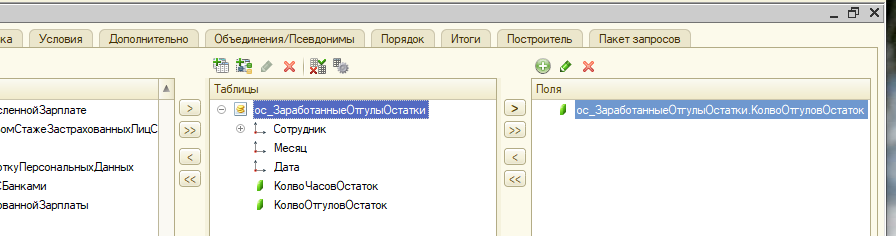


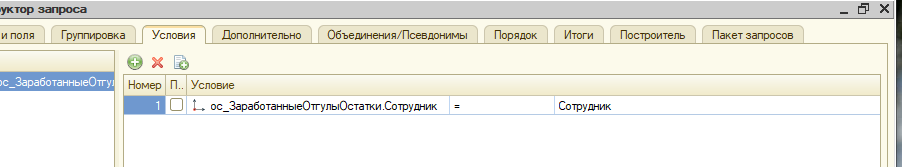


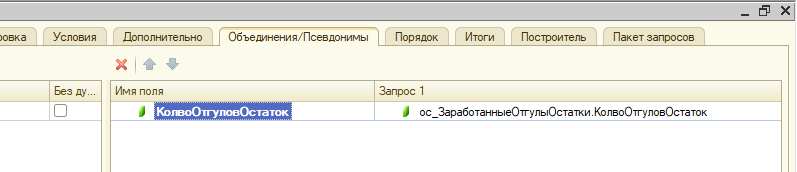




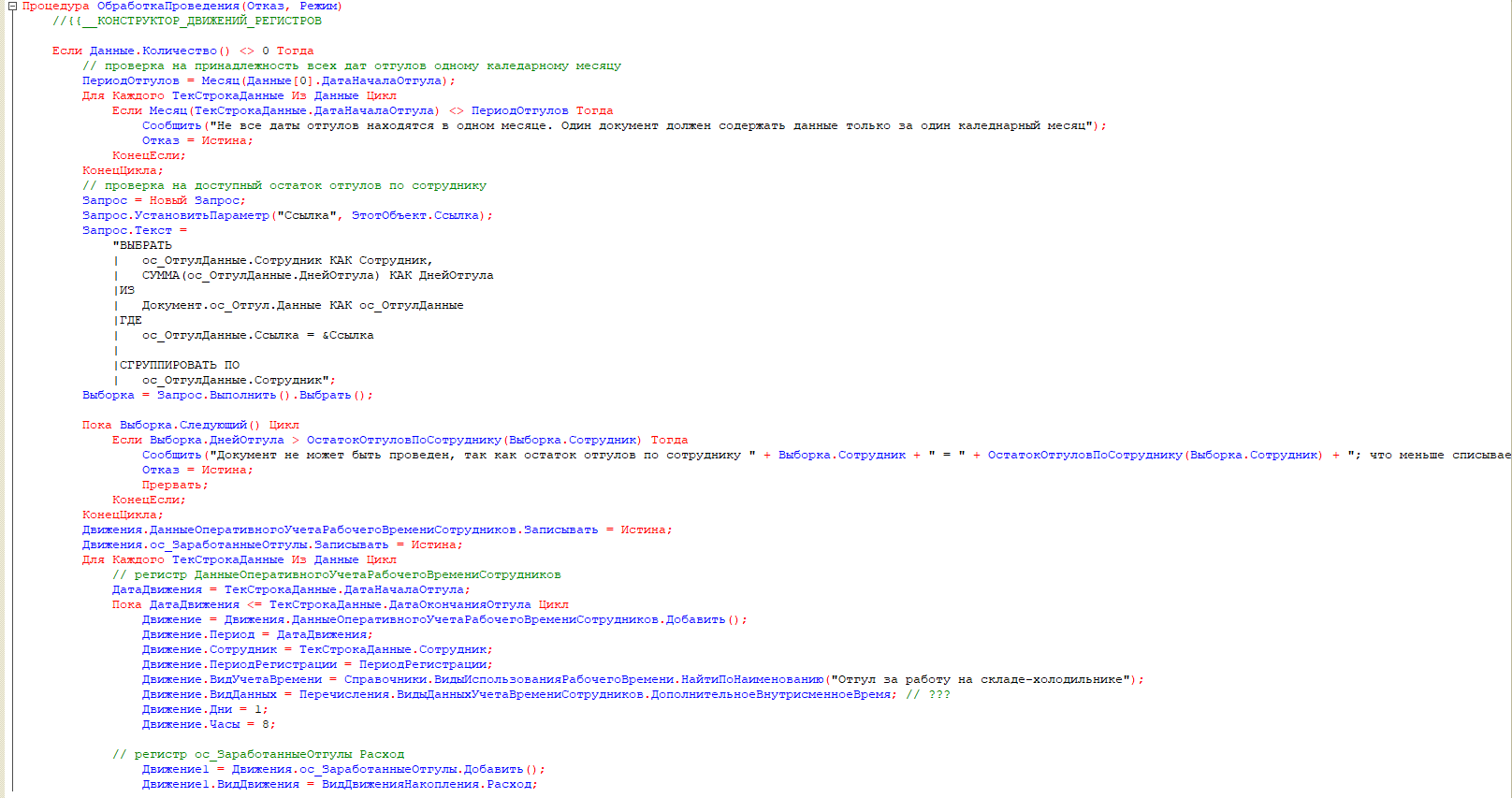
**ВТОРОЙ ЗАПРОС:**

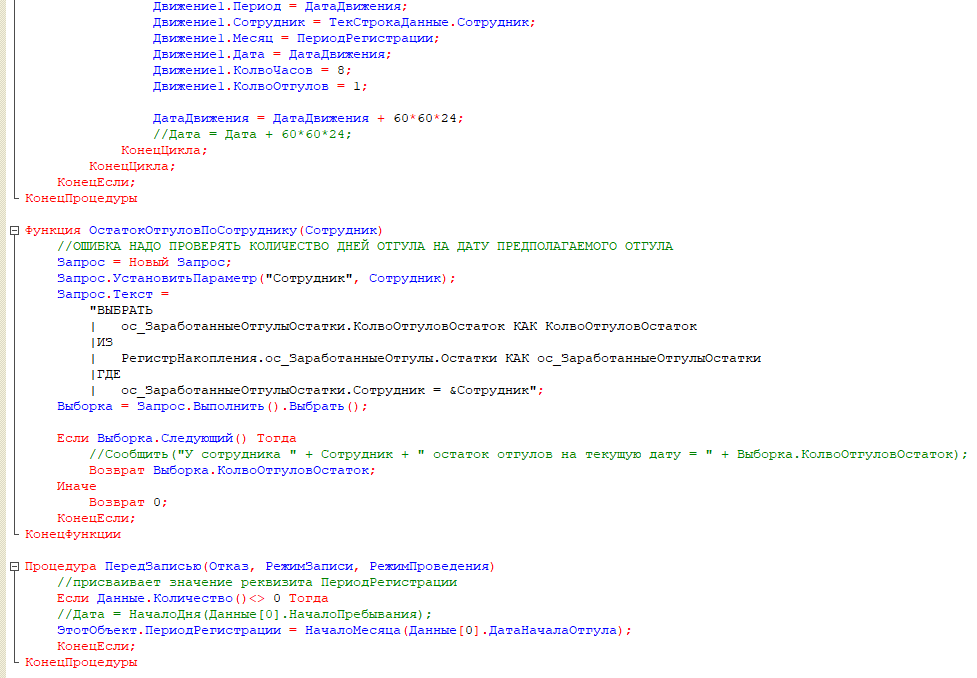






**МОДУЛЬ ДОКУМЕНТА ос\_Отгул ПОЛНОСТЬЮ:**





Процедура ОбработкаПроведения(Отказ, Режим)

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

Если Данные.Количество() <> 0 Тогда

// проверка на принадлежность всех дат отгулов одному каледарному месяцу

ПериодОтгулов = Месяц(Данные[0].ДатаНачалаОтгула);

Для Каждого ТекСтрокаДанные Из Данные Цикл

Если Месяц(ТекСтрокаДанные.ДатаНачалаОтгула) <> ПериодОтгулов Тогда

Сообщить("Не все даты отгулов находятся в одном месяце. Один документ должен содержать данные только за один каледнарный месяц");

Отказ = Истина;

КонецЕсли;

КонецЦикла;

// проверка на доступный остаток отгулов по сотруднику

Запрос = Новый Запрос;

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

Запрос.Текст =

"ВЫБРАТЬ

| ос\_ОтгулДанные.Сотрудник КАК Сотрудник,

| СУММА(ос\_ОтгулДанные.ДнейОтгула) КАК ДнейОтгула

|ИЗ

| Документ.ос\_Отгул.Данные КАК ос\_ОтгулДанные

|ГДЕ

| ос\_ОтгулДанные.Ссылка = &Ссылка

|

|СГРУППИРОВАТЬ ПО

| ос\_ОтгулДанные.Сотрудник";

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

Пока Выборка.Следующий() Цикл

Если Выборка.ДнейОтгула > ОстатокОтгуловПоСотруднику(Выборка.Сотрудник) Тогда

Сообщить("Документ не может быть проведен, так как остаток отгулов по сотруднику " + Выборка.Сотрудник + " = " + ОстатокОтгуловПоСотруднику(Выборка.Сотрудник) + "; что меньше списываемого количества, равного " + Выборка.ДнейОтгула);

Отказ = Истина;

Прервать;

КонецЕсли;

КонецЦикла;

Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Записывать = Истина;

Движения.ос\_ЗаработанныеОтгулы.Записывать = Истина;

Для Каждого ТекСтрокаДанные Из Данные Цикл

// регистр ДанныеОперативногоУчетаРабочегоВремениСотрудников

ДатаДвижения = ТекСтрокаДанные.ДатаНачалаОтгула;

Пока ДатаДвижения <= ТекСтрокаДанные.ДатаОкончанияОтгула Цикл

Движение = Движения.ДанныеОперативногоУчетаРабочегоВремениСотрудников.Добавить();

Движение.Период = ДатаДвижения;

Движение.Сотрудник = ТекСтрокаДанные.Сотрудник;

Движение.ПериодРегистрации = ПериодРегистрации;

Движение.ВидУчетаВремени = Справочники.ВидыИспользованияРабочегоВремени.НайтиПоНаименованию("Отгул за работу на складе-холодильнике");

Движение.ВидДанных = Перечисления.ВидыДанныхУчетаВремениСотрудников.ДополнительноеВнутрисменноеВремя; // ???

Движение.Дни = 1;

Движение.Часы = 8;

// регистр ос\_ЗаработанныеОтгулы Расход

Движение1 = Движения.ос\_ЗаработанныеОтгулы.Добавить();

Движение1.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение1.Период = ДатаДвижения;

Движение1.Сотрудник = ТекСтрокаДанные.Сотрудник;

Движение1.Месяц = ПериодРегистрации;

Движение1.Дата = ДатаДвижения;

Движение1.КолвоЧасов = 8;

Движение1.КолвоОтгулов = 1;

ДатаДвижения = ДатаДвижения + 60\*60\*24;

//Дата = Дата + 60\*60\*24;

КонецЦикла;

КонецЦикла;

КонецЕсли;

КонецПроцедуры

Функция ОстатокОтгуловПоСотруднику(Сотрудник)

//ОШИБКА НАДО ПРОВЕРЯТЬ КОЛИЧЕСТВО ДНЕЙ ОТГУЛА НА ДАТУ ПРЕДПОЛАГАЕМОГО ОТГУЛА

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Сотрудник", Сотрудник);

Запрос.Текст =

"ВЫБРАТЬ

| ос\_ЗаработанныеОтгулыОстатки.КолвоОтгуловОстаток КАК КолвоОтгуловОстаток

|ИЗ

| РегистрНакопления.ос\_ЗаработанныеОтгулы.Остатки КАК ос\_ЗаработанныеОтгулыОстатки

|ГДЕ

| ос\_ЗаработанныеОтгулыОстатки.Сотрудник = &Сотрудник";

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

Если Выборка.Следующий() Тогда

//Сообщить("У сотрудника " + Сотрудник + " остаток отгулов на текущую дату = " + Выборка.КолвоОтгуловОстаток);

Возврат Выборка.КолвоОтгуловОстаток;

Иначе

Возврат 0;

КонецЕсли;

КонецФункции

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

//присваивает значение реквизита ПериодРегистрации

Если Данные.Количество()<> 0 Тогда

//Дата = НачалоДня(Данные[0].НачалоПребывания);

ЭтотОбъект.ПериодРегистрации = НачалоМесяца(Данные[0].ДатаНачалаОтгула);

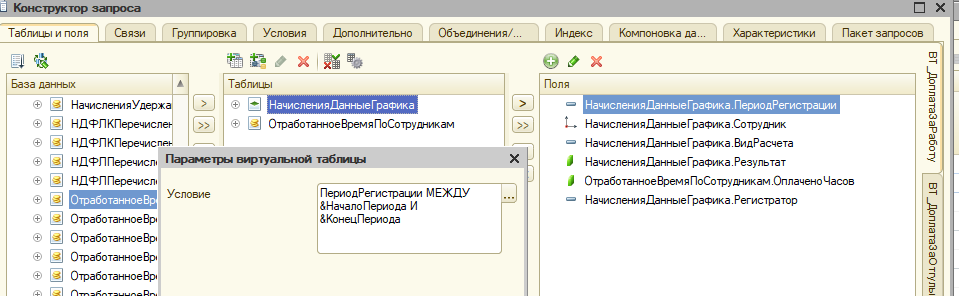
КонецЕсли;

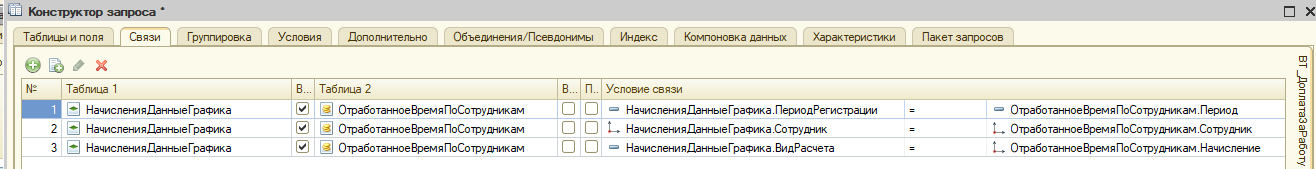
КонецПроцедуры

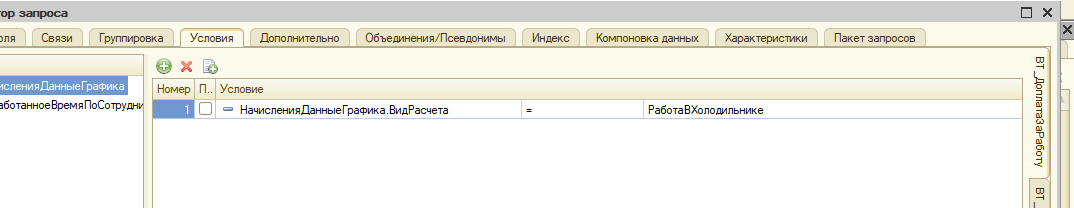
1. **Создаем отчет ос\_РаботыНаСкладе**

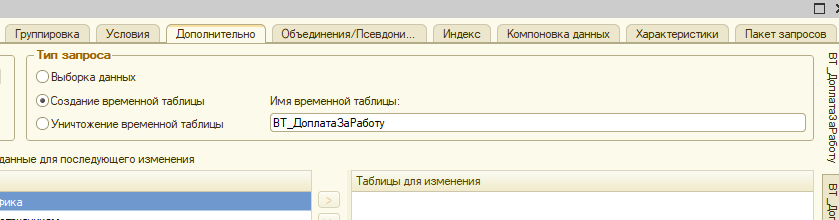
**ЗАПРОС:**

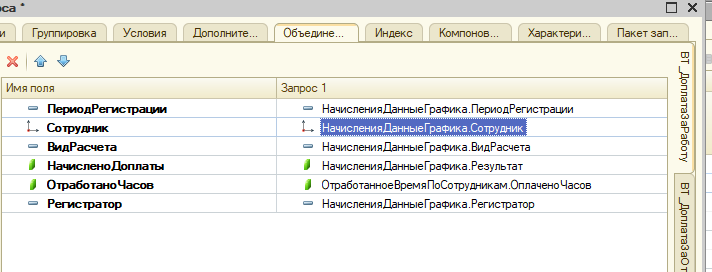
**ПЕРВЫЙ ПАКЕТ:**



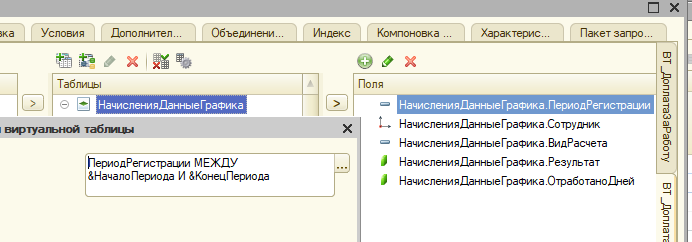


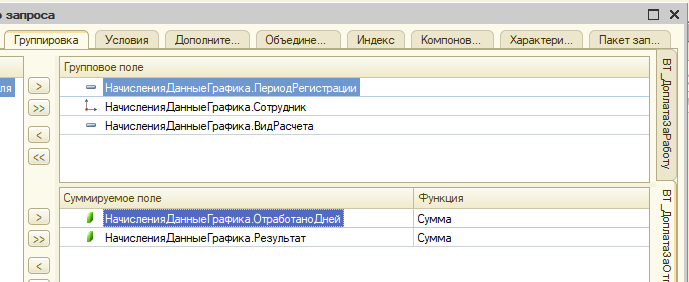


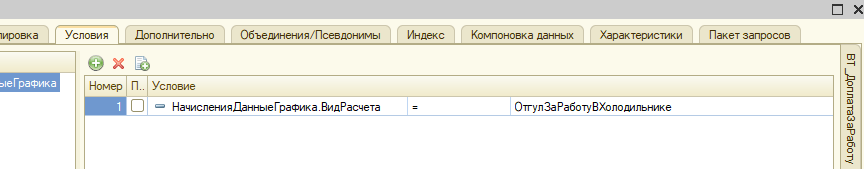


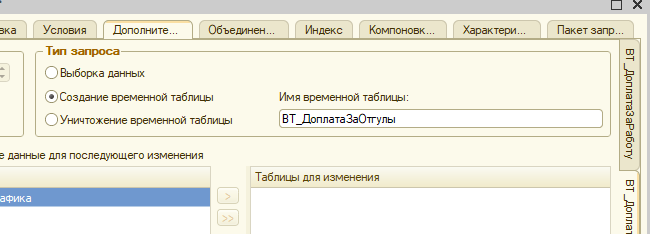


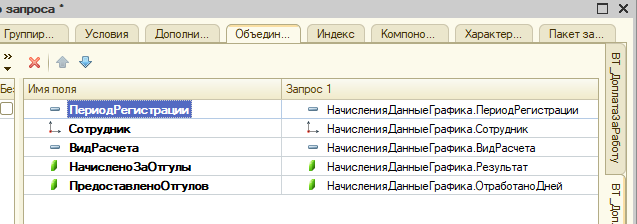
**ВТОРОЙ ПАКЕТ:**



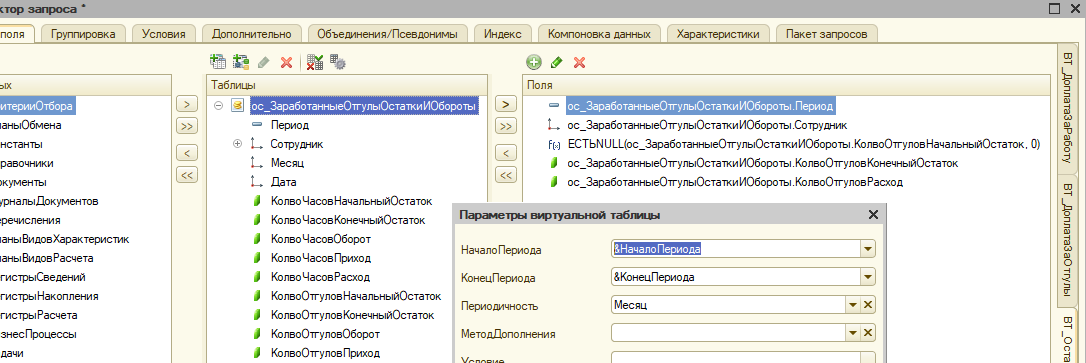


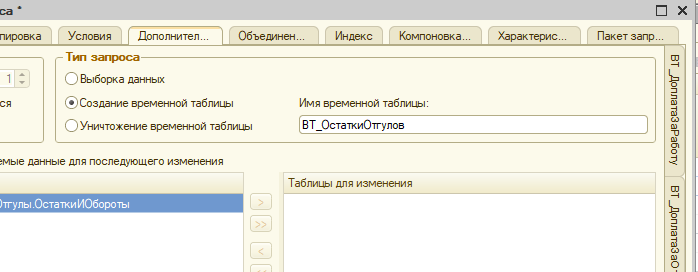


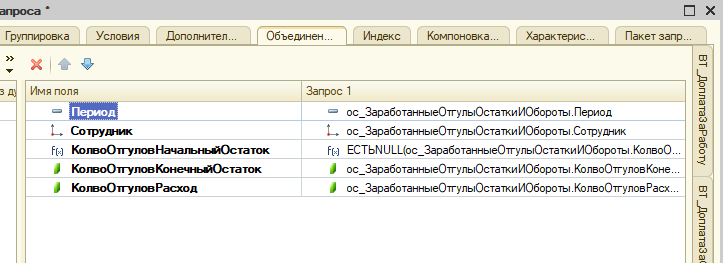




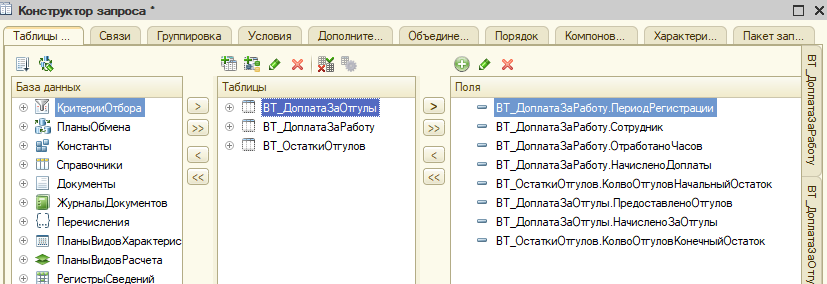
**ТРЕТИЙ ПАКЕТ:**

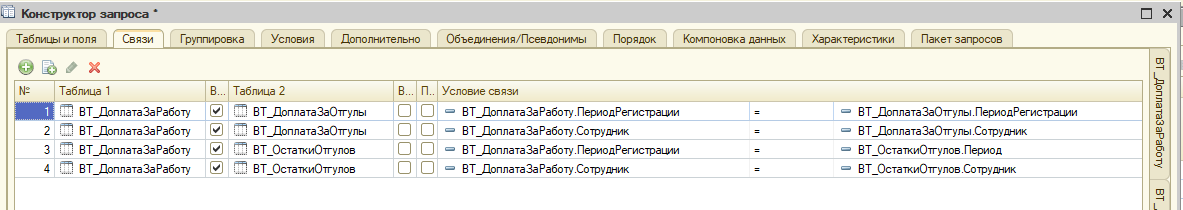


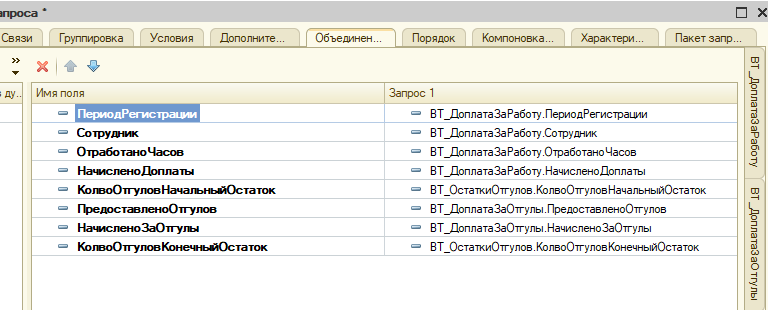




**ЧЕТВЕРТЫЙ ПАКЕТ:**







**Модуль запроса отчета:**

ВЫБРАТЬ

НачисленияДанныеГрафика.ПериодРегистрации КАК ПериодРегистрации,

НачисленияДанныеГрафика.Сотрудник КАК Сотрудник,

НачисленияДанныеГрафика.ВидРасчета КАК ВидРасчета,

НачисленияДанныеГрафика.Результат КАК НачисленоДоплаты,

ОтработанноеВремяПоСотрудникам.ОплаченоЧасов КАК ОтработаноЧасов,

НачисленияДанныеГрафика.Регистратор КАК Регистратор

ПОМЕСТИТЬ ВТ\_ДоплатаЗаРаботу

ИЗ

РегистрРасчета.Начисления.ДанныеГрафика(ПериодРегистрации МЕЖДУ &НачалоПериода И &КонецПериода) КАК НачисленияДанныеГрафика

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОтработанноеВремяПоСотрудникам КАК ОтработанноеВремяПоСотрудникам

ПО НачисленияДанныеГрафика.ПериодРегистрации = ОтработанноеВремяПоСотрудникам.Период

И НачисленияДанныеГрафика.Сотрудник = ОтработанноеВремяПоСотрудникам.Сотрудник

И НачисленияДанныеГрафика.ВидРасчета = ОтработанноеВремяПоСотрудникам.Начисление

ГДЕ

НачисленияДанныеГрафика.ВидРасчета = &РаботаВХолодильнике

;

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

ВЫБРАТЬ

НачисленияДанныеГрафика.ПериодРегистрации КАК ПериодРегистрации,

НачисленияДанныеГрафика.Сотрудник КАК Сотрудник,

НачисленияДанныеГрафика.ВидРасчета КАК ВидРасчета,

СУММА(НачисленияДанныеГрафика.Результат) КАК НачисленоЗаОтгулы,

СУММА(НачисленияДанныеГрафика.ОтработаноДней) КАК ПредоставленоОтгулов

ПОМЕСТИТЬ ВТ\_ДоплатаЗаОтгулы

ИЗ

РегистрРасчета.Начисления.ДанныеГрафика(ПериодРегистрации МЕЖДУ &НачалоПериода И &КонецПериода) КАК НачисленияДанныеГрафика

ГДЕ

НачисленияДанныеГрафика.ВидРасчета = &ОтгулЗаРаботуВХолодильнике

СГРУППИРОВАТЬ ПО

НачисленияДанныеГрафика.ПериодРегистрации,

НачисленияДанныеГрафика.Сотрудник,

НачисленияДанныеГрафика.ВидРасчета

;

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

ВЫБРАТЬ

ос\_ЗаработанныеОтгулыОстаткиИОбороты.Период КАК Период,

ос\_ЗаработанныеОтгулыОстаткиИОбороты.Сотрудник КАК Сотрудник,

ЕСТЬNULL(ос\_ЗаработанныеОтгулыОстаткиИОбороты.КолвоОтгуловНачальныйОстаток, 0) КАК КолвоОтгуловНачальныйОстаток,

ос\_ЗаработанныеОтгулыОстаткиИОбороты.КолвоОтгуловКонечныйОстаток КАК КолвоОтгуловКонечныйОстаток,

ос\_ЗаработанныеОтгулыОстаткиИОбороты.КолвоОтгуловРасход КАК КолвоОтгуловРасход

ПОМЕСТИТЬ ВТ\_ОстаткиОтгулов

ИЗ

РегистрНакопления.ос\_ЗаработанныеОтгулы.ОстаткиИОбороты(&НачалоПериода, &КонецПериода, Месяц, , ) КАК ос\_ЗаработанныеОтгулыОстаткиИОбороты

;

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

ВЫБРАТЬ

ВТ\_ДоплатаЗаРаботу.ПериодРегистрации КАК ПериодРегистрации,

ВТ\_ДоплатаЗаРаботу.Сотрудник КАК Сотрудник,

ВТ\_ДоплатаЗаРаботу.ОтработаноЧасов КАК ОтработаноЧасов,

ВТ\_ДоплатаЗаРаботу.НачисленоДоплаты КАК НачисленоДоплаты,

ВТ\_ОстаткиОтгулов.КолвоОтгуловНачальныйОстаток КАК КолвоОтгуловНачальныйОстаток,

ВТ\_ДоплатаЗаОтгулы.ПредоставленоОтгулов КАК ПредоставленоОтгулов,

ВТ\_ДоплатаЗаОтгулы.НачисленоЗаОтгулы КАК НачисленоЗаОтгулы,

ВТ\_ОстаткиОтгулов.КолвоОтгуловКонечныйОстаток КАК КолвоОтгуловКонечныйОстаток

ИЗ

ВТ\_ДоплатаЗаРаботу КАК ВТ\_ДоплатаЗаРаботу

ЛЕВОЕ СОЕДИНЕНИЕ ВТ\_ДоплатаЗаОтгулы КАК ВТ\_ДоплатаЗаОтгулы

ПО ВТ\_ДоплатаЗаРаботу.ПериодРегистрации = ВТ\_ДоплатаЗаОтгулы.ПериодРегистрации

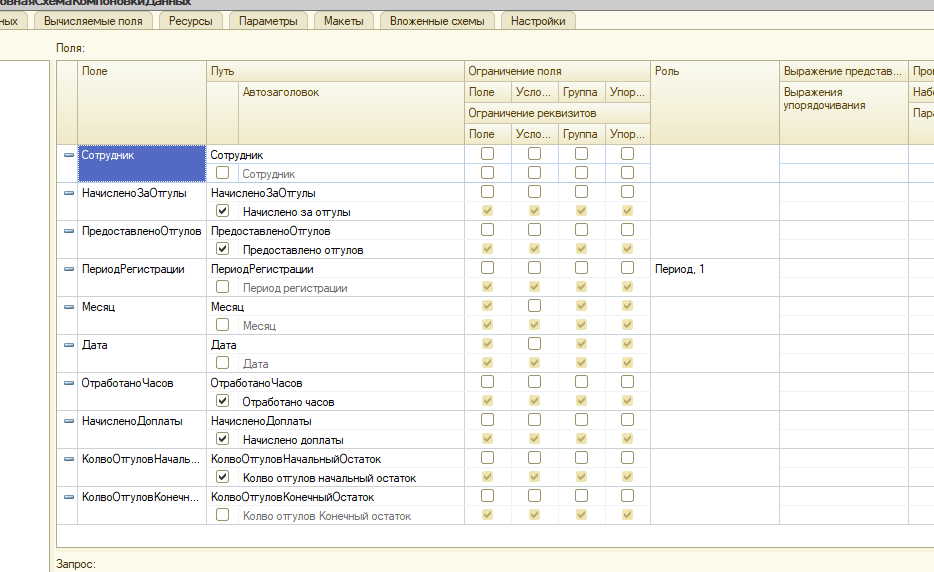
И ВТ\_ДоплатаЗаРаботу.Сотрудник = ВТ\_ДоплатаЗаОтгулы.Сотрудник

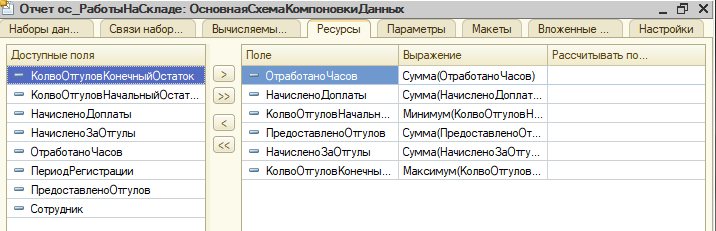
ЛЕВОЕ СОЕДИНЕНИЕ ВТ\_ОстаткиОтгулов КАК ВТ\_ОстаткиОтгулов

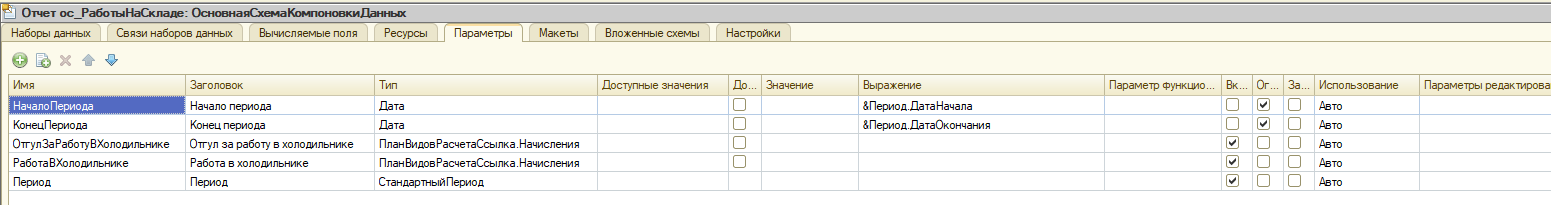
ПО ВТ\_ДоплатаЗаРаботу.ПериодРегистрации = ВТ\_ОстаткиОтгулов.Период

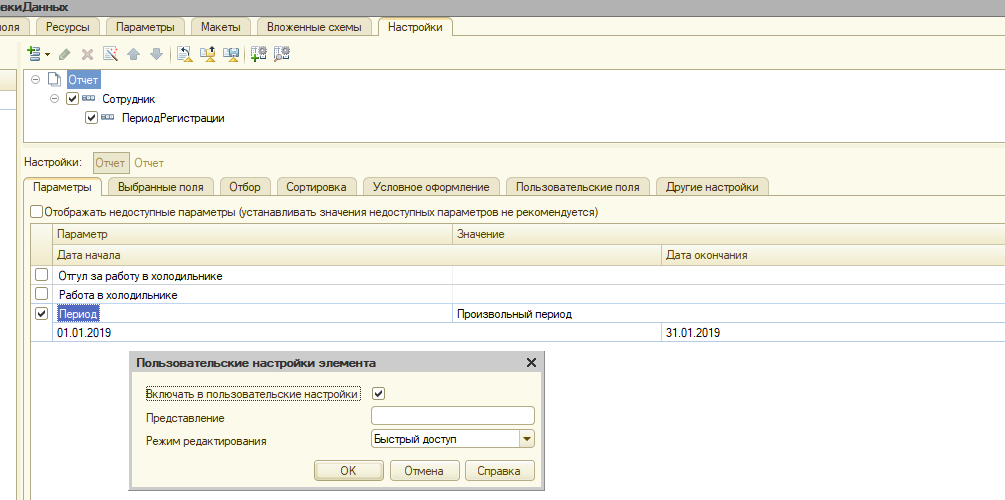
И ВТ\_ДоплатаЗаРаботу.Сотрудник = ВТ\_ОстаткиОтгулов.Сотрудник

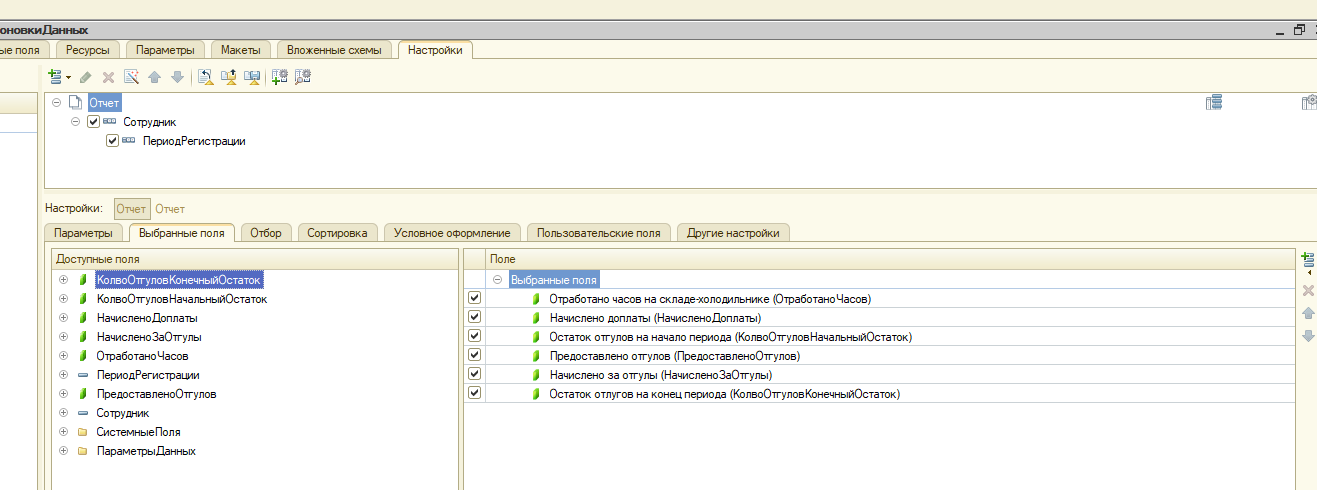
**Далее отчет в СКД:**

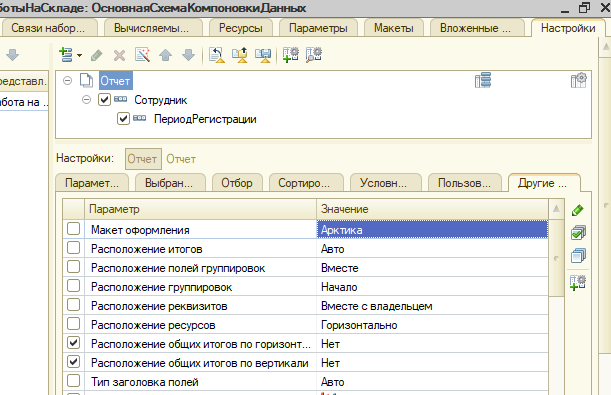






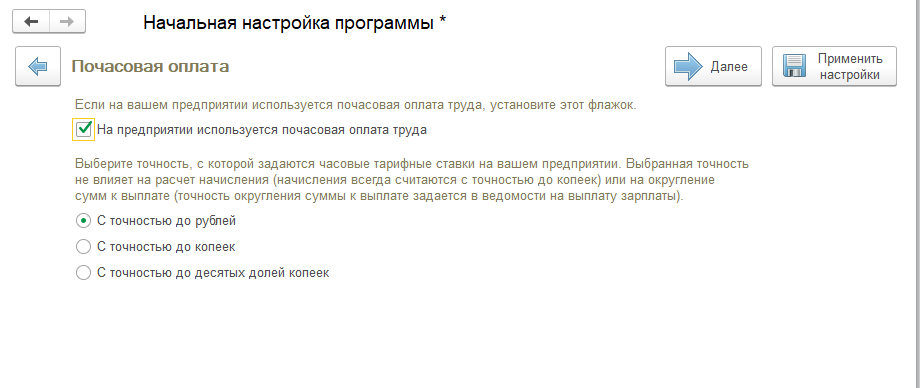




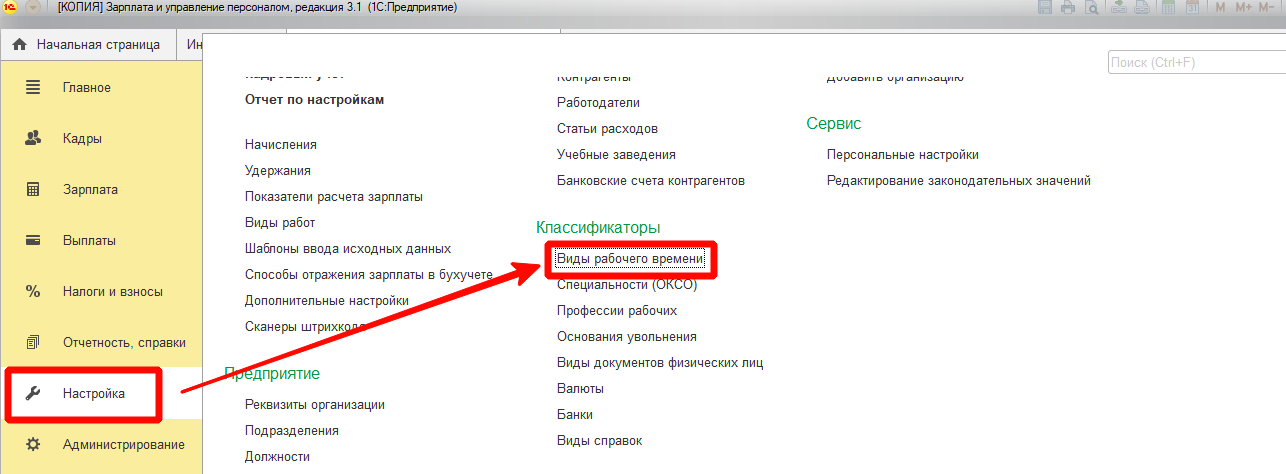


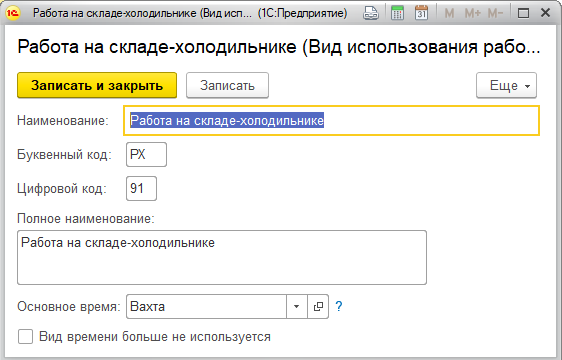
**III Ввод контрольного примера и проверка решения:**

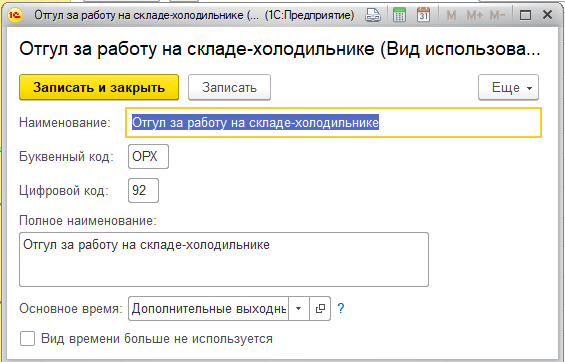
1. **Включаем почасовую оплату:**



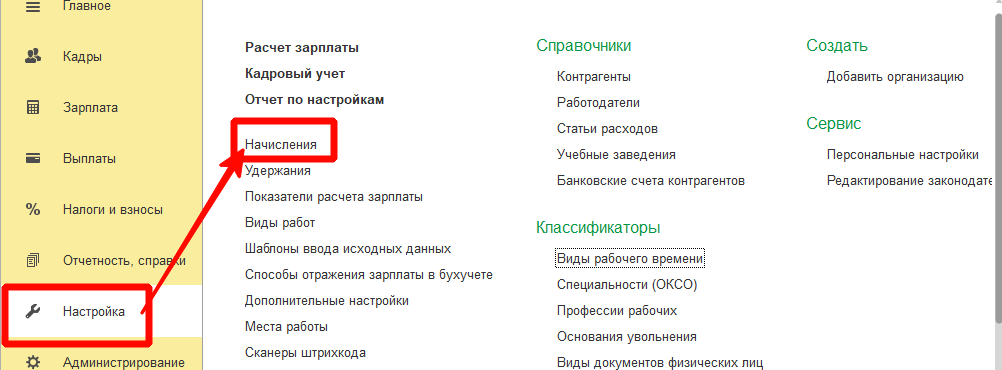
1. **Создаем виды рабочего времени:**



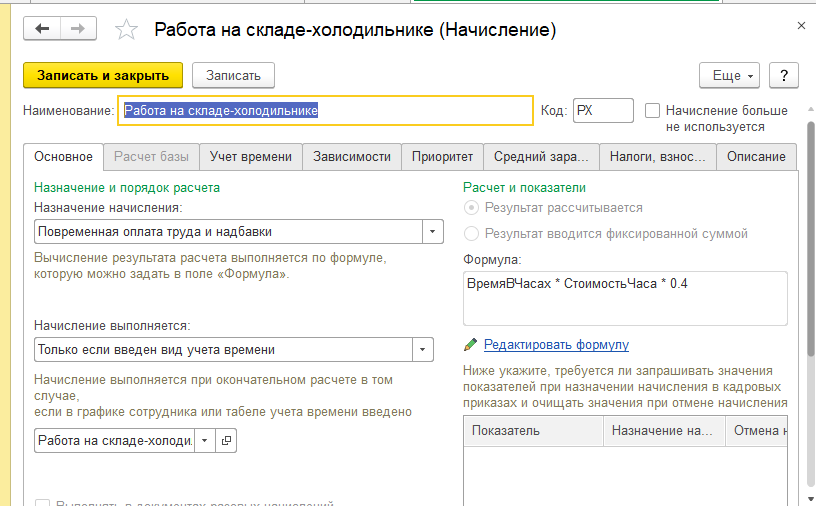




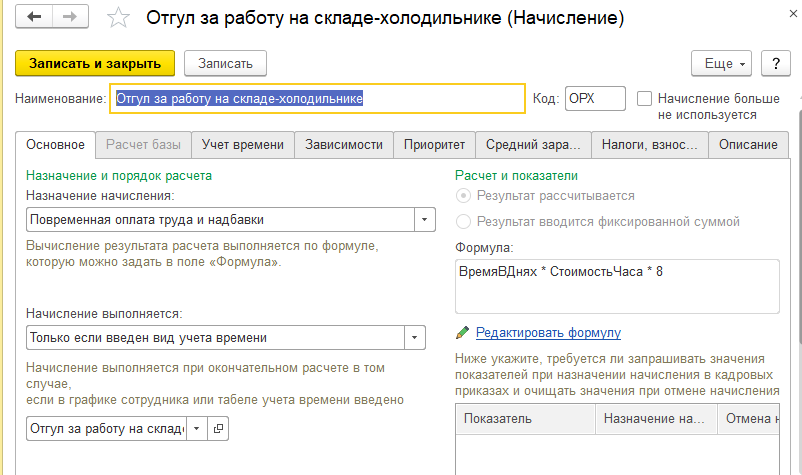
1. **Создаем два начисления:**

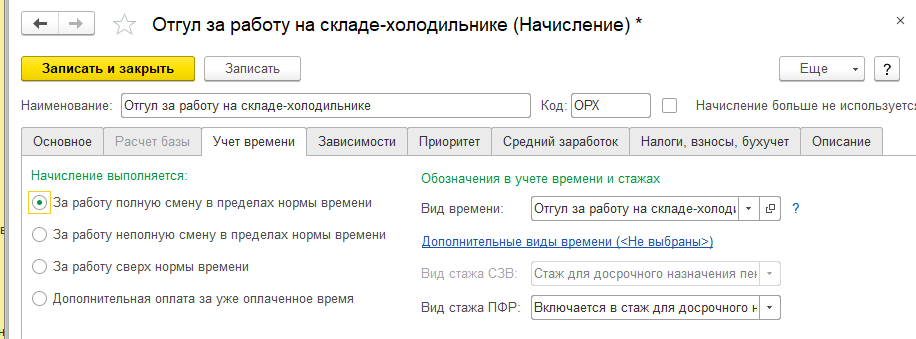


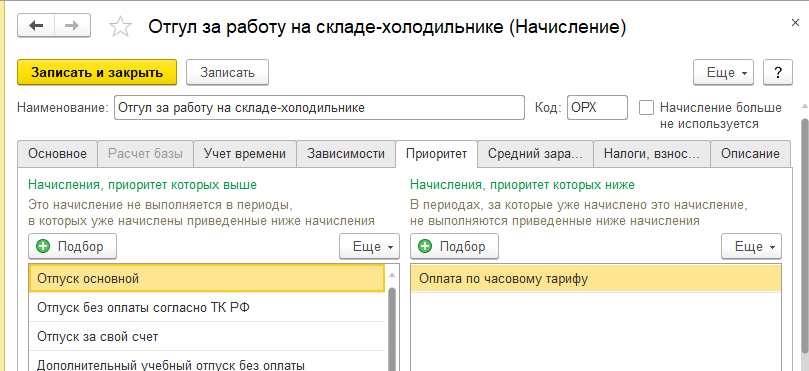
**Первое:**

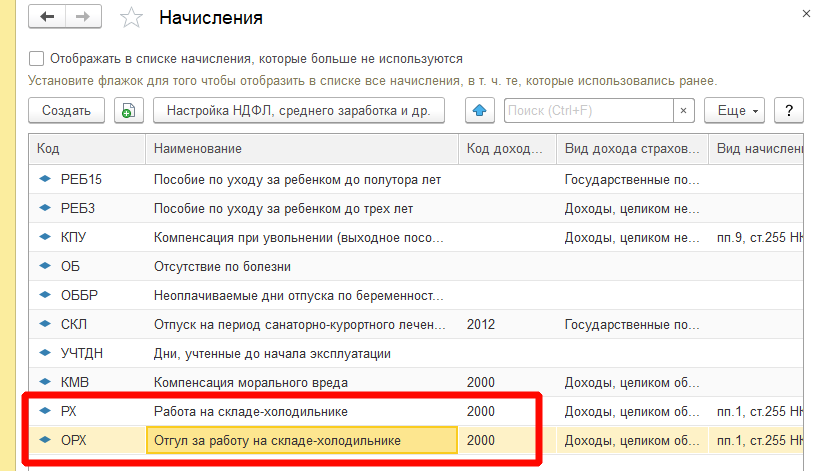


**Второе:**

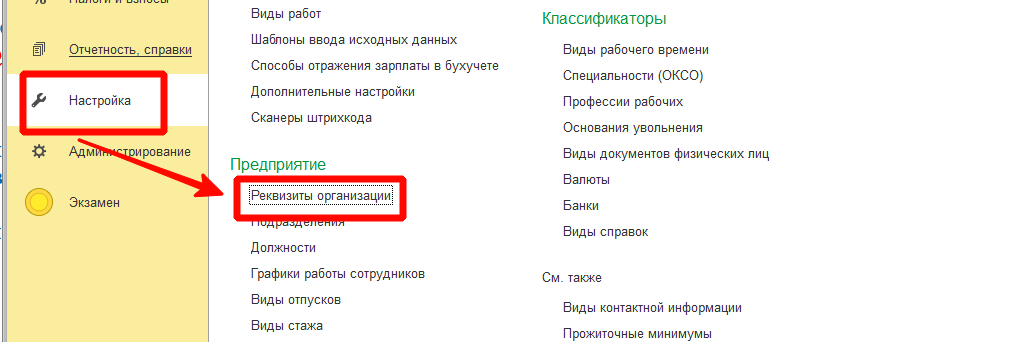


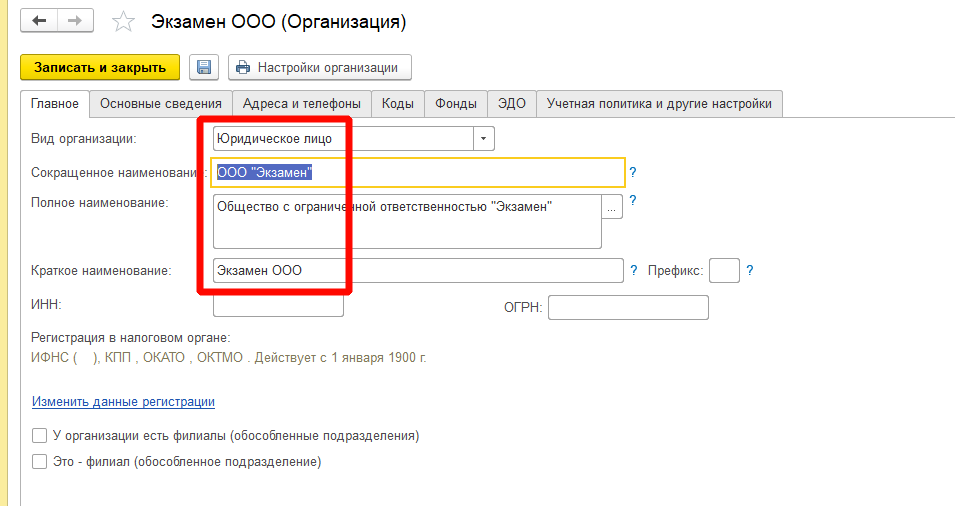




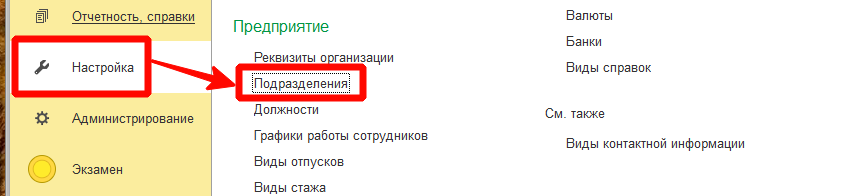


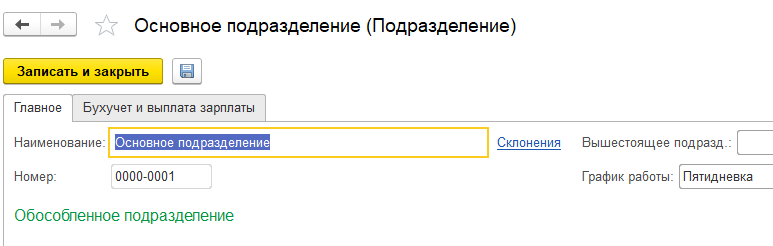
1. **Заполняем реквизиты организации:**



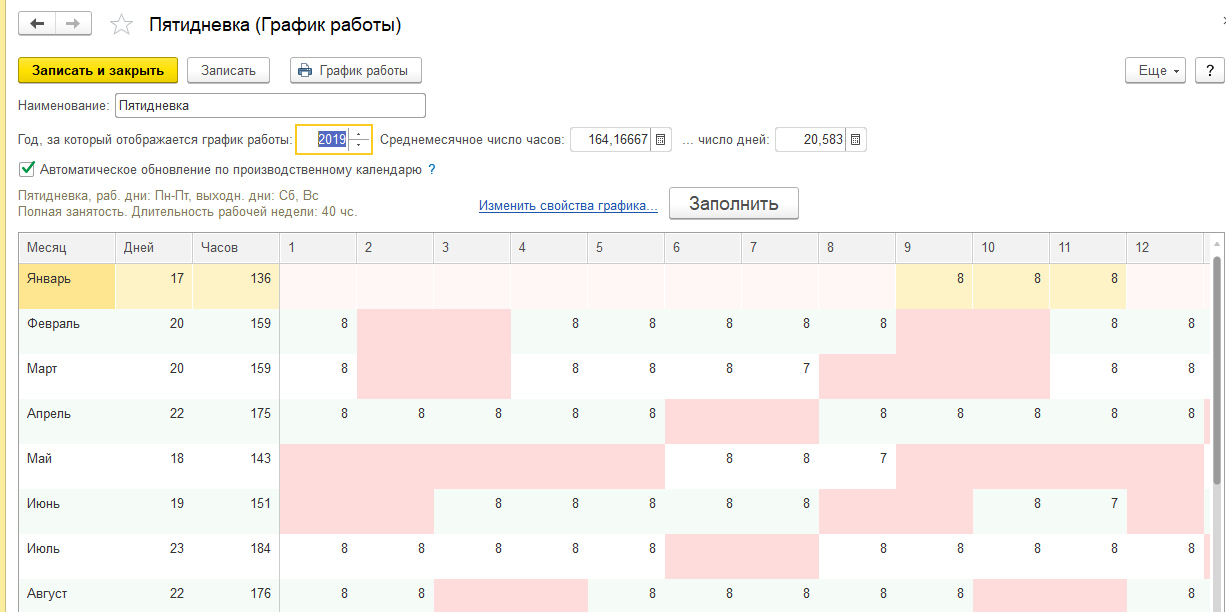


1. **Заполняем подразделение (с 2018 года):**

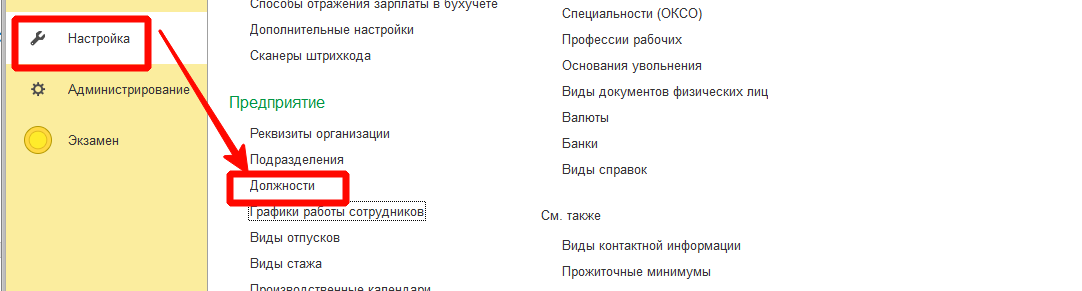


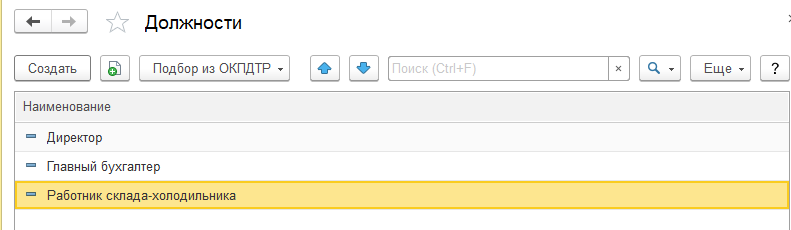


**В основном подразделении график «пятидневка», заполняем на 2019 год:**

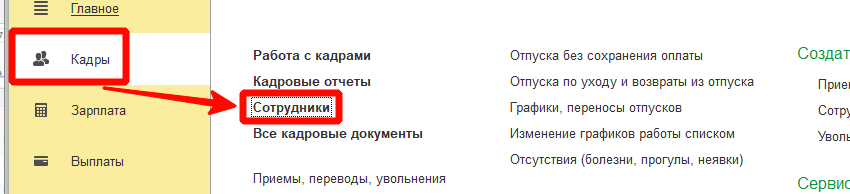


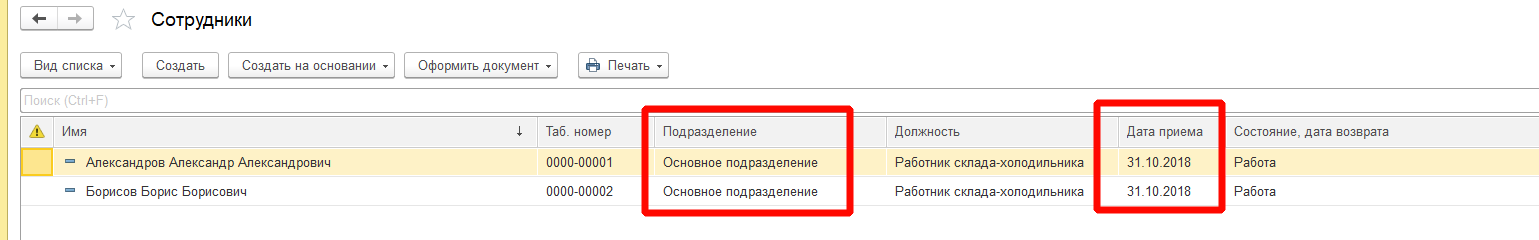
1. **Создаем должности как в задании:**

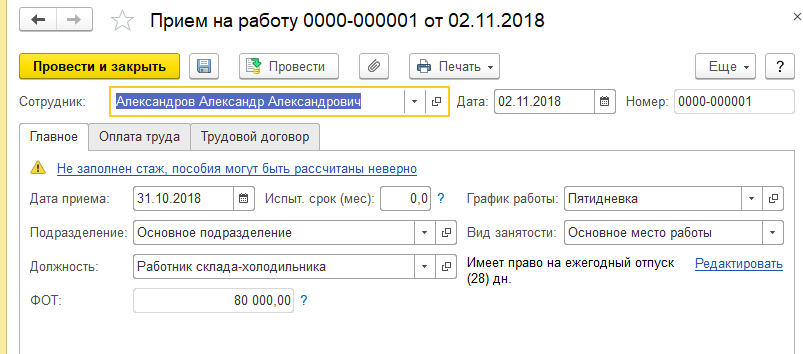


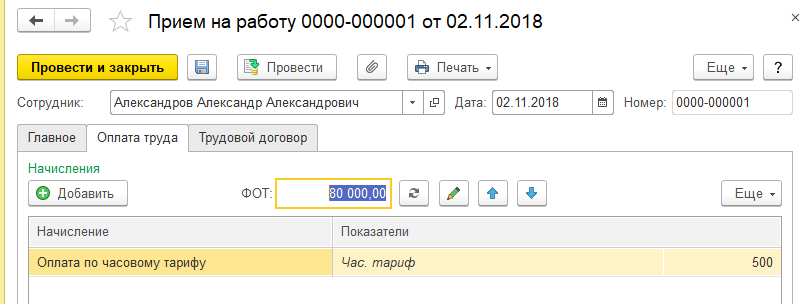


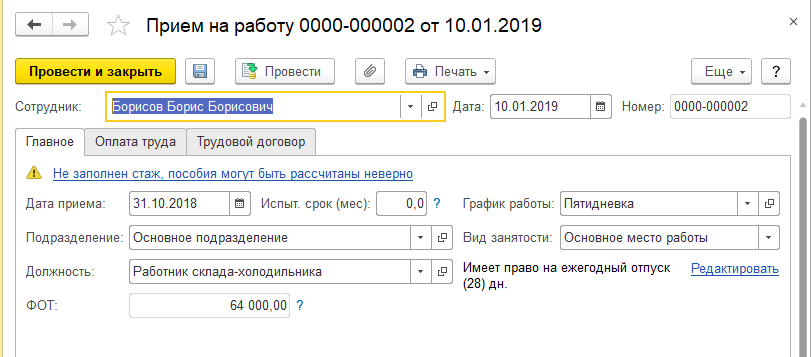
1. **Принимаем на работу двух сотрудников:**

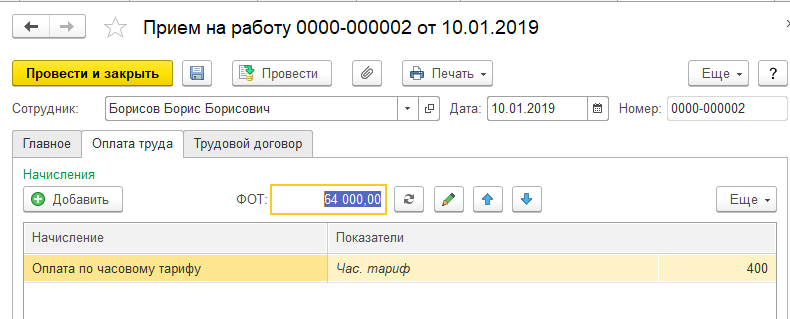


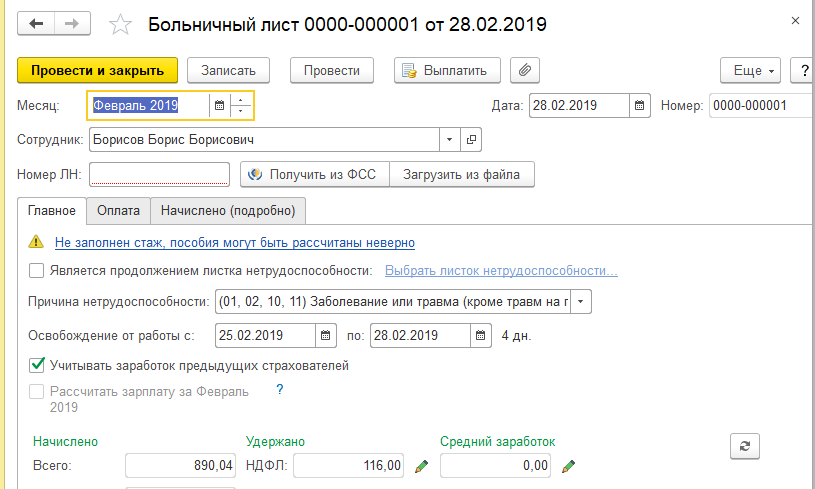




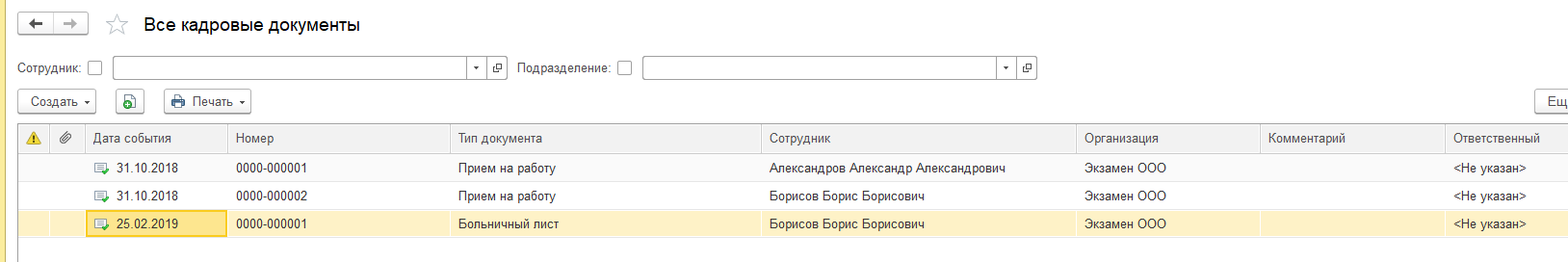




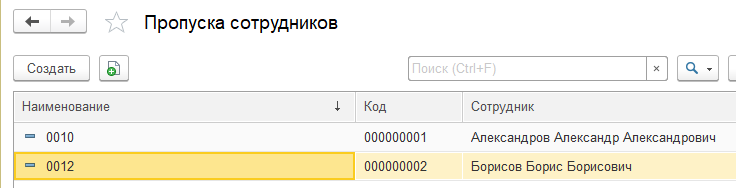




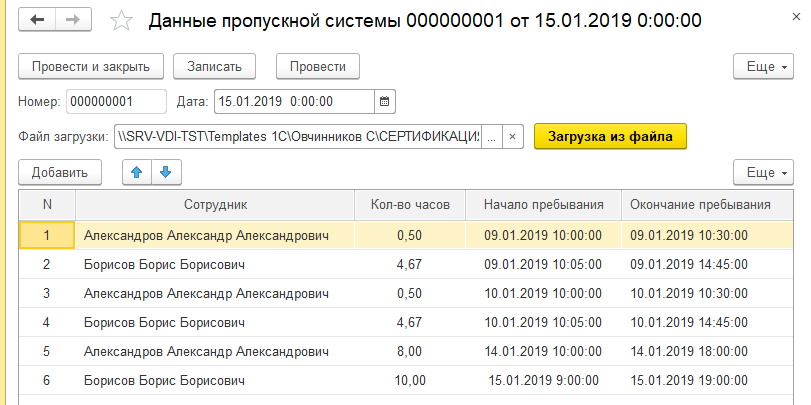
**Все кадровые документы:**

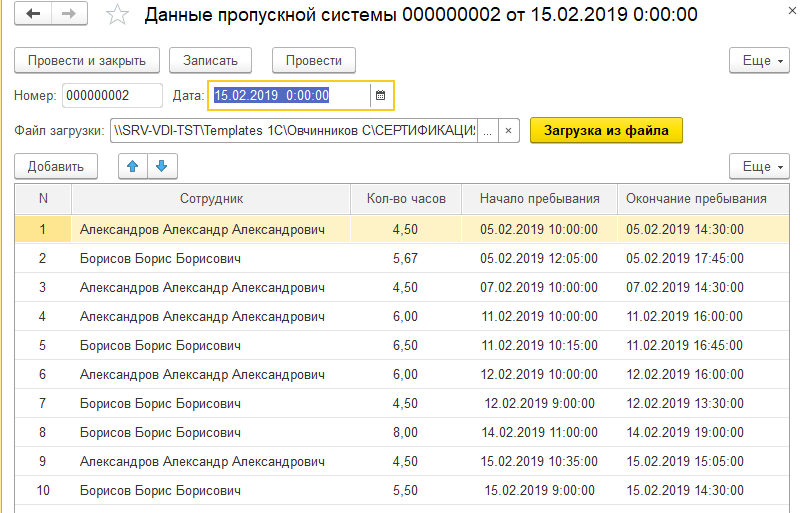


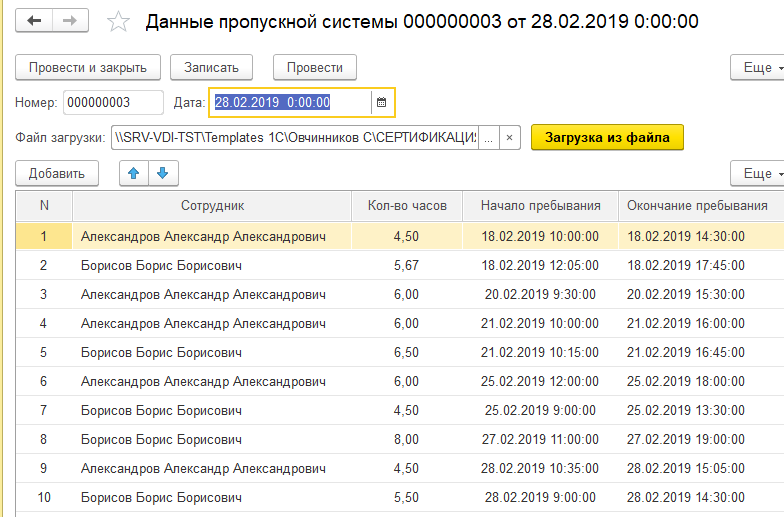
1. **Заполняем справочник Пропуска Сотрудников:**



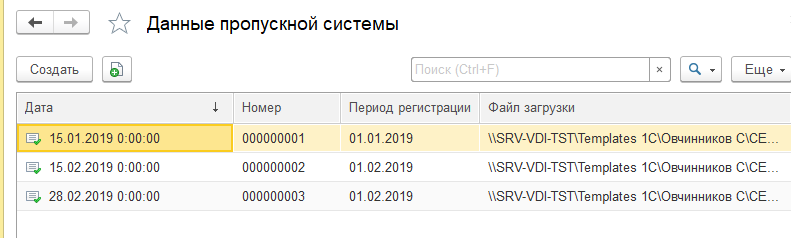
1. **Заполняем Данные пропускной системы (в точности как на скринах!!!!!!!!!!!):**



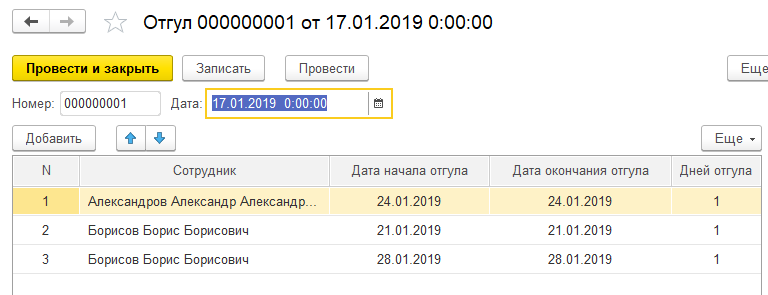


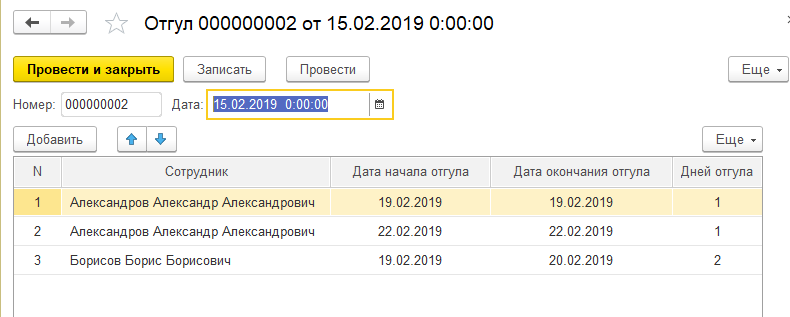


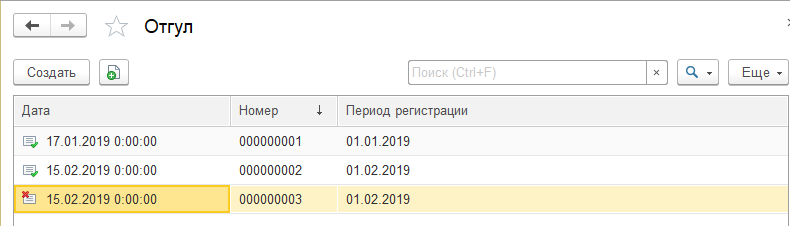
**Все документы пропускной системы:**



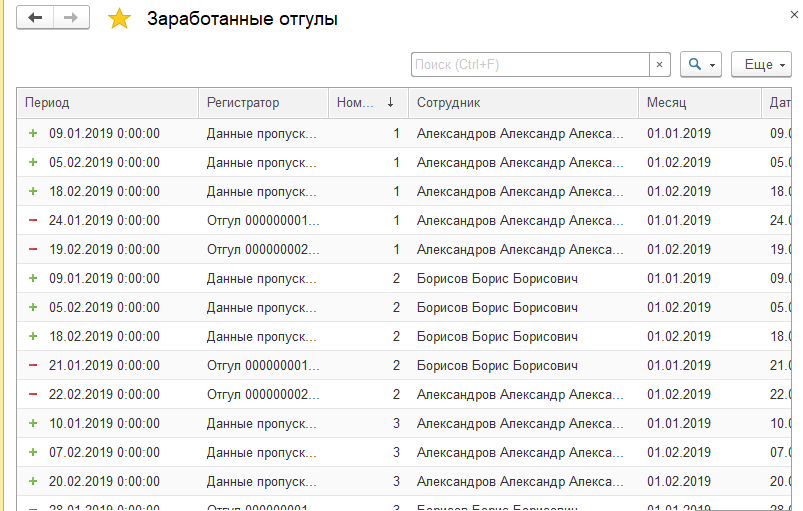
1. **Создаем отгулы:**



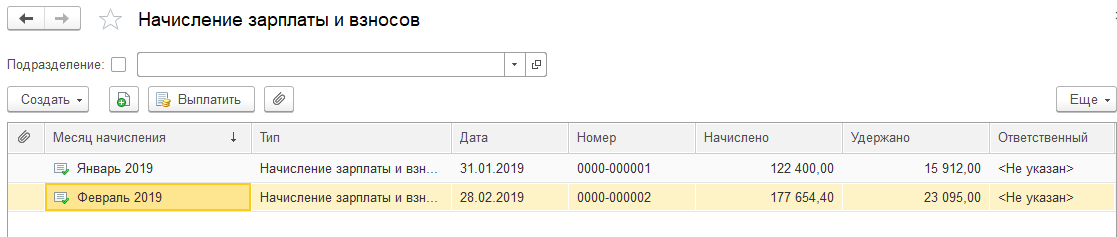




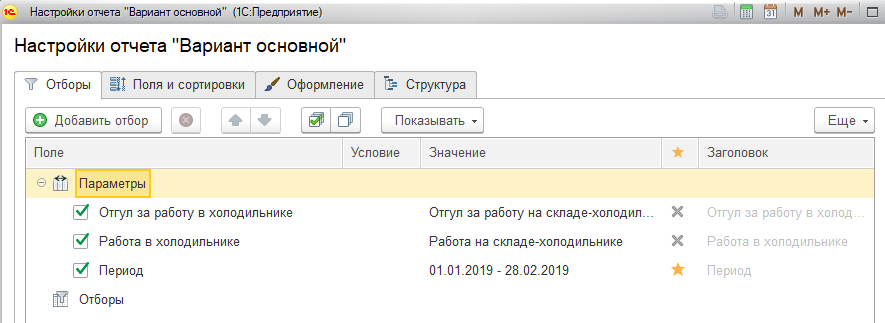
1. **Проверяем движения в РН Заработанные отгулы:**

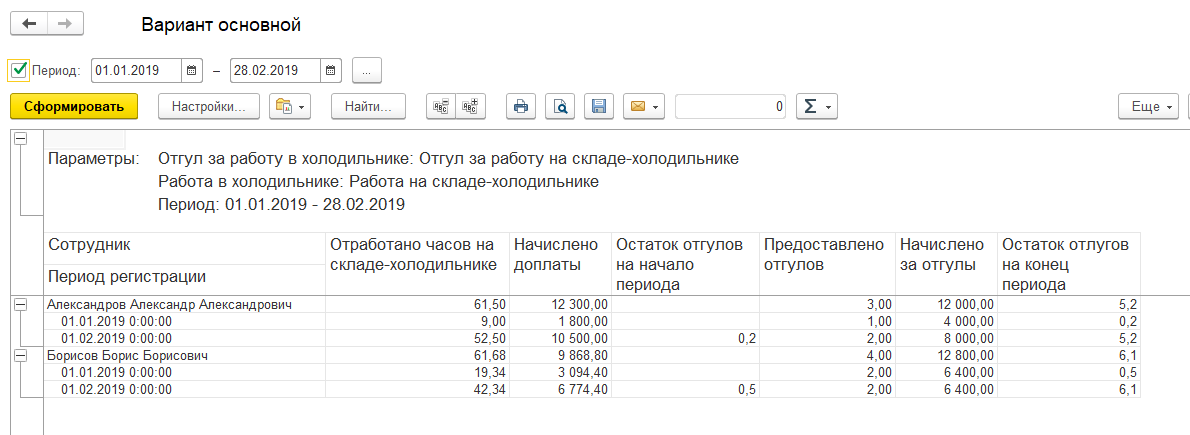


1. **Начисляем ЗП за два месяца:**



1. **Смотрим отчет:**





**IV ПОЯСНЕНИЕ К РЕШЕНИЮ**

Для реализации решения создал виды времени Работа на складе – холодильнике и Отгул за работу на складе - холодильнике и соответствующие начисления с формулами расчета согласно заданию. При формировании начислений в документе расчета зарплаты, если у начисления есть вытеснения приоритет которых ниже, начисление формируется именно за те дни, которые указаны в РС «ГрафикиРаботыПоВидамВремени» для сотрудника.

Для хранения заработанных отгулов создал РН (остатки) "ЗаработанныеОтгулы". В регистре храню и количество заработанных отгулов, и количество отработанных часов в холодильнике.

Разработан документ "ДанныеПропускнойСистемы", в модуле формы реализовал процедуру загрузки в табличную часть "Данные" из txt файла по описанному в задачи формату.

Документ формирует движения в РН "ДанныеОперативногоУчетаРабочегоВремениСотрудников" для учета начислений для целей налогообложения и обложения страховыми взносами в соответствии с законодательством и в РН "ЗаработанныеОтгулы" для учета отработанного времени.

Разработан документ "ос\_Отгул". Документ формирует движения в РН "ДанныеОперативногоУчетаРабочегоВремениСотрудников" и "ЗаработанныеОтгулы" для регистрации.

Разработал настраиваемый отчет "РаботыНаСкладе" в соответствии с предложенным в задании образцом.

Представленная к защите база содержит данные за два месяца по двум сотрудникам. Все начисления проведены и отображены в отчете. Зарплата начислена. Отчет проверен на работоспособность.

Таким образом, все требования задания выполнены.

**V ТЕКСТОВЫЕ ФАЙЛЫ**

|  |  |
| --- | --- |
| ТемпературыВоздухаЗаФевраль v1.0 | ТемпературыВоздухаЗаЯнварь v1.0 |
| 01.02.2019;-20  02.02.2019;2  03.02.2019;-3  04.02.2019;-15  05.02.2019;-25  06.02.2019;+3  07.02.2019;21  08.02.2019;-13  09.02.2019;-25  10.02.2019;35  11.02.2019;-20  12.02.2019;2  13.02.2019;-3  14.02.2019;-15  15.02.2019;-25  16.02.2019;+3  17.02.2019;21  18.02.2019;-13  19.02.2019;-25  20.02.2019;35  21.02.2019;-20  22.02.2019;2  23.02.2019;-3  24.02.2019;-15  25.02.2019;-25  26.02.2019;+3  27.02.2019;21  28.02.2019;-13 | 01.01.2019;-20  02.01.2019;2  03.01.2019;-3  04.01.2019;-15  05.01.2019;-25  06.01.2019;+3  07.01.2019;21  08.01.2019;-13  09.01.2019;-25  10.01.2019;35  11.01.2019;-20  12.01.2019;2  13.01.2019;-3  14.01.2019;-15  15.01.2019;-25  16.01.2019;+3  17.01.2019;21  18.01.2019;-13  19.01.2019;-25  20.01.2019;35  21.01.2019;-20  22.01.2019;2  23.01.2019;-3  24.01.2019;-15  25.01.2019;-25  26.01.2019;+3  27.01.2019;21  28.01.2019;-13  29.01.2019;-25  30.01.2019;35  31.01.2019;37 |