[Общее (ОУ,БУ,ПР) 3](#_Toc464822049)

[1) Проверка результата запроса на пустоту: 3](#_Toc464822050)

[2) Вопрос запрета оперативного проведения: 3](#_Toc464822051)

[3) Свойство «Удаление движений»: 4](#_Toc464822052)

[4)Режим разделения итогов: 4](#_Toc464822053)

[5)Получение данных за период: 4](#_Toc464822054)

[6) Флаг Общего модуля "Вызов сервера": 4](#_Toc464822055)

[7) Обращение к реквизиту составного типа в запросе: 4](#_Toc464822056)

[8) Основные задачи, решаемые при реализации количественного учета: 5](#_Toc464822057)

[9)Оптимальное проведение документа списания товаров при реализации количественного учета 5](#_Toc464822058)

[10) «Защита от дурака» за счёт установки свойств элементов интерфейса: 5](#_Toc464822059)

[11) Привязка сообщения к элементу форму: 5](#_Toc464822060)

[12) Рекомендации по установке управляемой блокировки при использовании старой методики проведения: 6](#_Toc464822061)

[13) Новая технология проведения предполагает следующий порядок действий при реализации количественного учета: 6](#_Toc464822062)

[14) Использование объектной модели "РегистрНакопления.Менеджер" 6](#_Toc464822063)

[15) Составление схемы запроса выполняется по следующей методике для новичков: 8](#_Toc464822064)

[16) Произвольное условие в конструкторе запросов: 9](#_Toc464822065)

[17) Правила выполнения запросов: 9](#_Toc464822066)

[18) Управляемые блокировки: 10](#_Toc464822067)

[19) Правила внесения изменений в структуру регистров рабочей базы: 10](#_Toc464822068)

[20) Направления оптимизации алгоритма проведения: 11](#_Toc464822069)

[21)Оптимизация оборотных регистров. Агрегаты: 12](#_Toc464822070)

[БУ 13](#_Toc464822071)

[1)Ручная операция, уточнения по кор.счету: 13](#_Toc464822072)

[2) Флаг "Балансовый": 13](#_Toc464822073)

[3) Реализация аналитического учёта : 13](#_Toc464822074)

[4)Рекомендации по определению субконто: 14](#_Toc464822075)

[5) Индексирование таблиц регистра бухгалтерии: 14](#_Toc464822076)

[6) Анализ результата запроса, когда неизвестно на каком месте находится какое субконто: 15](#_Toc464822077)

[7)Получение развернутого сальдо из виртуальных таблиц: 15](#_Toc464822078)

[8) Количественный учет 15](#_Toc464822079)

[9) Валютный учет: 15](#_Toc464822080)

[10) Документ РучнаяОперация: 16](#_Toc464822081)

[11) Объект НаборЗаписей регистра бухгалтерии 18](#_Toc464822082)

[12) МодульНабораЗаписей регистра бухгалтерии 18](#_Toc464822083)

[13) Признаки учета субконто: 19](#_Toc464822084)

[ОУ: 20](#_Toc464822085)

[1) Корректировка записей регистров: 20](#_Toc464822086)

[2) Оптимизированный алгоритм проведения документа 21](#_Toc464822087)

[3) Правило упрощения реализации запроса по нескольким источникам 21](#_Toc464822088)

[4) Виртуальные таблицы 22](#_Toc464822089)

[5) Реализация партионного учета 23](#_Toc464822090)

[6)Организация планирования процесса оказания постпродажных услуг. Работа с регистром сведений. 23](#_Toc464822091)

[СКД 24](#_Toc464822092)

[ПР 24](#_Toc464822093)

[1) Период регистрации: 24](#_Toc464822094)

[2) Состав расчетных механизмов платформы: 25](#_Toc464822095)

[3) Настройка расчетных свойств плана видов расчета: 26](#_Toc464822096)

[4) Выбор критерия попадания в базовый период: 27](#_Toc464822097)

[5) Вытесняющие виды расчета: 28](#_Toc464822098)

[6) Регистры расчета: 28](#_Toc464822099)

[7) Настройка Измерений, Ресурсов и Реквизитов РР: 29](#_Toc464822100)

[8) Этапы сеанса расчета зарплаты: 30](#_Toc464822101)

[9) Особенности метода Записать набора записей РР 31](#_Toc464822102)

[10) Виртуальная таблица РР ДанныеГрафика: 32](#_Toc464822103)

[11) Подход к установлению соответствия строк регистра и строки необходимых данных 33](#_Toc464822104)

[12) Виртуальная таблица РР ФактическийПериодДействия 34](#_Toc464822105)

[13) Виртуальная таблица База РР 34](#_Toc464822106)

[14) Параметры виртуальной таблицы База 35](#_Toc464822107)

[15) Состав полей виртуальной таблицы База : 36](#_Toc464822108)

[16) Использование периода действия в качестве базового периода 36](#_Toc464822109)

[17) Расчет суммы начисления , зависящей от нескольких базовых начислений 36](#_Toc464822110)

[18) Частичное попадание периода действия базового вида расчета в базовый период 37](#_Toc464822111)

[19) Получение суммы базы в разрезе базовых ВР: 38](#_Toc464822112)

[20) Построения отчета по начислениям. 38](#_Toc464822113)

[21) Корректировка результатов расчета за прошедший период 38](#_Toc464822114)

[22)Перерасчеты 40](#_Toc464822115)

[23) Объект конфигурации Перерасчет 40](#_Toc464822116)

[24) Объектная модель работы с перерасчетом 41](#_Toc464822117)

[25) Две особенности реализации Перерасчетов 42](#_Toc464822118)

[26) Расчет зарплаты по табелю 42](#_Toc464822119)

[26.1) Виды расчета: 44](#_Toc464822120)

[26.2) Расчет зарплаты: 46](#_Toc464822121)

[26.3) Отчет: 50](#_Toc464822122)

[27)Перетекающий период: 50](#_Toc464822123)

[28) МногократноеИзменениеОклада: 53](#_Toc464822124)

# Общее (ОУ,БУ,ПР)

## 1) Проверка результата запроса на пустоту:

ЗначениеЗаполнено для Результат запроса неверно использовать, т.к. есть моменты когда это отработает некорректно. Рекомендуется использовать Результат.Пустой()

## 2) Вопрос запрета оперативного проведения:

Насчет оперативного проведения: нигде не прописано, что его не нужно обрабатывать, и в курсах 1С по подготовке к экзамену этого нет -там преподаватели обрабатывают оперативное проведение.

Тем не менее его можно не обрабатывать, и даже запретить, т.к. это упростило бы жизнь.   
Вот сообщение, где сказано (в самом конце), что за это делали замечания:   
[http://forum.chistov.pro/index.php?topic=993.msg47005..](https://vk.com/away.php?to=http%3A%2F%2Fforum.chistov.pro%2Findex.php%3Ftopic%3D993.msg47005%23msg47005)   
Дальше обсуждали - это замечание основано на ошибке из перечня: "Если при проведении документа используются каким-то образом данные, считываемые из регистров, обязательно требуется предусмотреть получение таких данных на момент проведения документа". Эта ошибка = -1 балл. При обработке оперативного проведения как раз получаются данные НЕ на момент проведения документа.   
По поводу запрета оперативного проведения - идея не нова, так делают много лет, тема поднималась в сообщениях 2011-2012 годов, насколько мне известно, за это замечаний не

Цитата 1С:Специалиста по платформе:

«

… Я сам на экз опер проведение запрещал во всех доках, но, по-моему, на это Гончаров просто не смотрел; в коде оперативное проведение я не обрабатывал, это Гончаров, естественно, видел, замечаний не было…

»

Оперативное проведение чаще всего не нужно при решении бухгалтерских задач.

Выдержка из курса "Решение задач бух.учета", стр.45   
"...  
Свойство "Оперативное проведение" устанавливаем в значение "Запретить". Контроль оперативности имеет смысл только для учета в реальном времени и в бухгалтерском учете совсем или почти не используется  
..."

## 3) Свойство «Удаление движений»:

Свойство "Удаление движений" устанавливаем в значение "Удалять автоматически при отмене проведения" - это стандартный вариант. Это позволит перезаписывать набор записей регистра при перепроведении документа без предварительной очистки существующих. Существующие будут удаляться автоматически, только если документ снимается с проведения (удаляется).  
  
В обработчике проведения необходимо сначала записывать пустой набор в базу данных. В свойствах документа свойство "Удалять движения" по умолчанию установлено в значение "Удалять автоматически при отмене проведения", поэтому при перепроведении документа его старые проводки будут находиться в наборе движений, иначе говоря будут доступны для считывания, потому их и нужно таким образом очистить. Если пользователь при перепроведении не изменит время документа на более позднее, тогда ошибки не будет, поскольку остатки считываются на момент времени документа , т.е. его проводки не включатся. Если пользователь не записывает набор пустых движений и изменит время проведения документа на более позднее или проведение будет оперативным, возникнет ошибка, т.к. движения документа, записанные до перепроведения будут учтены, что будет ошибкой.

## 4)Режим разделения итогов:

Режим разделения итогов включен по умолчанию для регистров конфигурации

## 5)Получение данных за период:

При получении данных ЗА ПЕРИОД следует брать конец периода даты окончания периода:  
- СКД КонецПериода(&ИмяПериода,"День");  
- Параметр запроса КонецДня(ИмяПараметра)  
  
При получении остатка НА ДАТУ(конец дня, включая его последнюю секунду) следует воспользоваться объектом Граница:  
  
Момент = Новый Граница(КонецДня(ИмяПараметра),ВидГраницы.Включая);  
  
, где ИмяПараметра - параметр, содержащий значение даты окончания

## 6) Флаг Общего модуля "Вызов сервера":

Флаг Общего модуля "Вызов сервера" необходимо установить, если вызов экспортных методов данного модуля(выполняемых на сервере), вызывается из области, выполняемой на Клиенте.

## 7) Обращение к реквизиту составного типа в запросе:

В случае осуществления запроса к реквизиту значения составного типа необходимо использовать предложение ВЫРАЗИТЬ, например:  
  
ВЫРАЗИТЬ ( ОстаткиИОбороты.Субконто1 КАК Справочник.Номенклатура) КАК Субконто1

## 8) Основные задачи, решаемые при реализации количественного учета:

1.Эффективное считывание данных из базы и документа, необходимых для проведения.  
2.Расчет стоимости списания, которую обычно рассчитывают по одной из принятых методик ( по средней, FIFO или LIFO):  
Окр(Выборка.СуммаОстаток \* Выборка.Количество / Выборка.КоличествоОстаток, 2)  
(видел в методичке 1с Решение задач БУ : стр.135)  
3.Контроль возможности проведения документа / контроль остатков  
4.Блокирование считываемых данных при многопользовательском режиме в клиент - серверной реализации.

## **9)Оптимальное проведение документа списания товаров при реализации количественного учета**

Запрос использует временные таблицы и состоит из нескольких пакетов:  
  
1.1. Первый пакет запроса получает сгруппированные данные табличной части документа с отбором по данному документу и необходимыми условиями (например отбор только Товаров) с индексацией по номенклатуре. Результат выполнения помещён во временную таблицу. По данным этой временной таблицы имеет смысл блокировать записи таблицы РН, в случае использования старой методики проведения, т.е. тогда, когда реализовать списание товара из регистра невозможно без получения данных из регистра. Например, для списания себестоимости необходимо знать количество и стоимость остатков списываемой номенклатуры.

1.2. Второй пакет запроса получает остатки с отбором по номенклатуре из первого пакета, значения ресурсов проверены на отсутствие значения (NULL), данные из временной таблицы соединены левым соединением с виртуальной таблицей остатков (при необходимости сгруппированы по номенклатуре (если это партионный учет, отсортированы по моменту времени Партии))

## **10) «Защита от дурака» за счёт установки свойств элементов интерфейса:**

Следует помнить о возможности установки ограничений в полях ввода в формах, а именно где необходимо – запрещать незаполненные значения, устанавливать маски, устанавливать диапазон допустимых значений. В большинстве случаев это решает вопрос «защиты от дурака» на начальном уровне.

11) Привязка сообщения к элементу форму:  
Прикрепление сообщения о нехватке товара при списании к столбцу Количество ТЧ Товары. Рекомендуется делать во всех случаях для доведения приема до автоматизма(одна из задач на УФ):  
"...  
Сообщение. Текст = " Не хватает" + Выборка.Представление + " в количестве " + Число(Выборка.Количество - Выборка.КоличествоОстаток) ;  
Сообщение. Поле = "Товары[" + (Выборка.НомерСтроки -1) + "].Количество";  
Сообщение.УстановитьДанные(ЭтотОбъект);  
  
..."

## **12) Рекомендации по установке управляемой блокировки при использовании старой методики проведения:**

1. Следует устанавливать блокировку максимально близко к концу транзакции (чем меньше блокировка длится, тем больше пользователей работает одновременно, т.е. выше параллельность).  
2. Известно, какие данные блокировать  
3.Блокировка должна производиться до обращения к конкурентному ресурсу, например, виртуальной таблице РегистраБухгалтерии.  
  
БУ:  
  
Уточняющие условия можно установить с применением метода УстановитьЗначение для блокировки по значению. Для тех свойств блокировки, где в качестве параметра нужно передать список или массив значений, можно использовать свойств ИсточникДанных и метод ИспользоватьИзИсточникаДанных. В свойство ИсточникДанных можно поместить табличную часть документа, таблицу значений или результат запроса, а в методе вторым параметром указать имя колонки, значения и которой используются для фильтра.  
  
Для субконто есть специфика установки значения блокировки - не всегда точно известно на какой позиции закреплено искомое субконто, поэтому используется ссылка на вид характеристики (вид субконто) для обозначения поля пространства блокировки.

## **13) Новая технология проведения предполагает следующий порядок действий при реализации количественного учета:**

1.Описываем формирование движений документа, установив для них специальный флаг блокирования данных итогов (БлокироватьПриИзменении).   
  
2.Выполняем запись набора движений в регистр, при этом на данные итогов, соответствующие значениям измерений, используемых в наборе, накладывается блокировка.  
  
3.После записи и установки блокировки выполняем контроль остатков, при этом ищем отрицательные . При наличии отрицательных остатков отменяем проведение и выводим уведомление пользователю.

## **14) Использование объектной модели "РегистрНакопления.Менеджер"**

В большинстве случаев получение данных в виде таблицы значений считается менее эффективным. По этой причине объектная модель получения данных считается менее эффективной нежели табличная. Например, результат выполнения метода Остатки объекта РегистрНакопленияМенеджер есть таблица значений, заполненная итогами, содержащая колонки с ресурсами, указанными в параметре Ресурсы.

Рассмотрим данную технологию на примере. Необходимо дать пользователю возможность контролировать остатки в старых документах на дату их проведения. Для этого необходимо показать на форме документа остатки товаров, попавших в табличную часть, актуальные на момент регистрации самого документа. Эта задача реализована следующим образом:

1.В форму документа добавлен реквизит Остаток(Строка).

2. Поскольку необходимо обработать событие изменения табличной части документа, код будет располагаться в обработчике события "ПриАктивизацииСтроки" форму документа. В обработчике пропишем следующий код:

"...

&НаКлиенте

Процедура ТоварыПриАктивизацииСтроки(Элемент)

Стр = Элементы.Товары.ТекущиеДанные;

Если Стр <> Неопределено Тогда

Ост = ОпределениеОстаткаТовара(Стр.Номенклатура,

Объект.Склад, Объект.Дата, Объект.Ссылка);

КонецЕсли;

КонецПроцедуры

&НаСервереБезКонтекста

Функция ОпределениеОстаткаТовара (ТекТовар, Склад, Дата, Ссылка)

ОстаткиТов = РегистрыНакопления.ОстаткиНоменклатуры;

Фильтр = Новый Структура;

Фильтр.Вставить("Номенклатура", ТекТовар);

Фильтр.Вставить("Склад", Склад);

Если ЗначениеЗаполнено(Ссылка) Тогда

//если док. есть в базе, у него заполнены поля дата и ссылка, значит по ним можно

//сформировать МоментВремени для отбора остатков

ВременнойМомент = Новый МоментВремени(Дата, Ссылка);

Иначе

//получаем самые актуальные остатки

ВременнойМомент = Неопределено;

КонецЕсли;

ТаблицаОст = ОстаткиТов.Остатки(ВременнойМомент, Фильтр,"Номенклатура, Количество");

//возврат итога по колонке Количество

Возврат ТаблицаОст.Итог("Количество");

КонецФункции

..."

Объектная модель используется пока нужно получать итоги с отборами типа "равно определенному значению". Рассмотрим пример, когда применение объектной модели будет невозможно:

1. Нужно показать остатки по складу и фирме в целом;

2. Нужно получить остатки по всем товарам, попавшим в табличную часть документа;

3. нужно получить свободный остаток по товару

Все эти задачи не решить эффективно посредством объектной за одно обращение к БД.

Наиболее частые случаи использования объектной модели это:

1.Когда прикладной объект надо записать в БД

2.Когда прикладной объект надо модифицировать в БД (прочитать в оперативную память, изменить и записать измененный в БД).

3.Когда в алгоритме нужно использовать механизм динамического чтения данных больших массивов прикладных объектов ( суть механизма - чтение данных из БД блоками по небольшому количеству записей), можно использовать объект манипулирования данными <СправочникВыборка.<ИмяСправочника», <ДокументВыборка.<ИмяДокумента», <РегистрНакопленияВыборка.<ИмяРегистра» и т.п.

4.Когда данные уже находятся в оперативной памяти и к ним нужно обратиться.

В остальных случаях применение запроса для решения задачи или одинаково эффективно с объектной моделью или более эффективно.

## **15) Составление схемы запроса выполняется по следующей методике для новичков:**

Составление схемы запроса выполняется по следующей методике для новичков:

1.Нарисовать эскиз выходной таблицы запроса с вариантом заполнения данных;

2.Обозначить на эскизе количество выходных полей.

3. Обозначить для каждого поля таблицу - источник ( по возможности консолидируя разрезы)

4. Обозначить дополнительное действие в запросе.

К дополнительным действиям чаще всего относятся:

- фильтрация, отбор ( когда речь идёт об отборе по значению измерения виртуальной таблицы, тогда надо делать отбор в параметрах самой таблицы, в остальных случаях - отборы с предложением ГДЕ).

- свертка таблицы (СГРУППИРОВАТЬ ПО);

- разворачивание таблицы за счет добавления строк с промежуточными итогами ( ИТОГИ ПО);

- сортировка (УПОРЯДОЧИТЬ ПО).

## 16) Произвольное условие в конструкторе запросов:

Произвольное условие в конструкторе запросов (на вкладке "Условия") можно построить путём вложенного вызова конструктора произвольных выражений конструктора запросов. Для этого после установки флага "Произвольное" следует однократно нажать в правый угол поля, в котором располагается произвольное условие. Таким образом можно будет активировать кнопку выбора в конце строки. При нажатии на данную кнопку выбора будет вызван конструктор произвольных выражений.

## **17) Правила выполнения запросов:**

С точки зрения техники выполнения запроса желательно помнить следующие правила:

1. Соединение - попытка в одном запросе получить все возможные комбинации записей исходных таблиц источников, ограниченная количеством и видами связей. Связь - условие, которому обязательно соответствуют выбираемые комбинации.

Если связей не накладывать, тогда записи таблиц соединятся по схеме "каждый с каждым". Во избежание этого при написании запроса руководствуются правилом: "Минимальное количество связей на единицу меньше количества таблиц источников" - больше связей допустимо, меньше - нет.

2. В прикладных задачах чаще всего есть возможность определить ведущий и ведомый источники данных. Выходные поля группировочных сущностей должны определяться из ведущих источников.

При написании запроса, помимо СКД, важно помнить следующую закономерность. Когда в качестве источника выбрана виртуальная таблица, имеет смысл войти в свойства виртуальной таблицы и указать имена всем внешним параметрам, связанным со временем построения виртуальной таблицы.

В СКД эти параметры устанавливаются системой автоматически.

Когда нужно выбрать все данные, методически правильнее использовать параметр, который будет установлен в значение Неопределено или '00010101', что соответствует значению Неопределено для типа данных Дата.

При написании запроса, если в самом запросе необходимо наложить условие на виртуальную таблицу, и условие касается отбора по одному или нескольким значениям измерений, такое условие должно накладываться в параметрах виртуальной таблицы. Такое условие выполняется сразу при построении виртуальной таблицы. Если наложить условие по измерению после построения виртуальной таблицы, быстродействие запроса значительно снизится по понятным причинам.

Для повышения эффективности запроса имеет смысл накладывать условия на виртуальные таблицы по максимуму. Кроме того в запросах по полям отбора целесообразно строить индекс, чтобы повысить скорость построения виртуальной таблицы по заданному условию.

При оперативном проведении, чтобы документы располагались последовательно и не попадали внутрь одной секунды ( внутри секунды порядок следования не определен) система выполняет автоматическое изменение времени документа. Дата документа получает значение оперативной отметки времени, которая рассчитывается системой по следующему алгоритму: дата оперативно проводимого документа равна текущей или на секунду больше предыдущей выданной оперативной отметки времени. При оперативном проведении документов система позволяет добиться расстановки документов в правильном хронологическом порядке (каждое событие зарегистрировано на момент его отражения в учете) и, кроме того, известить об этом обработчик проведения ( с точки зрения отражения документа в учёте сам механизм оперативного проведения не выполняет действий, а передаёт параметр в обработчик проведения). Это даёт разработчику возможность отличать ситуации проведения документов реальным временем от прочих ситуаций, что используется для решения прикладных задач. Например, в случае оперативного проведения можно получить актуальные остатки, что гораздо быстрее, чем получение остатков на произвольный момент времени.

3. Объединение - это добавление снизу к выходной таблице запроса по первому источнику выходных таблиц по последующим источникам, значит, в случае Объединения для каждого источника нужен свой запрос. Это значит, что колонки с группировками (например, номенклатурой) должны указываться в качестве выходных полей в каждом из этих запросов.

4. При реализации объединения важно, чтобы каждый из объединяемых запросов имел одинаковое количество полей. Если для одного из наборов поля объединения отсутствуют, отсутствующие поля следует дополнить пустыми значениями, чтобы сделать количество полей равным.

## **18) Управляемые блокировки:**

Управляемые блокировки:

Установка полей блокировки для единичных значений используется метод УстановитьЗначение объекта ЭлементБлокировки. Для установки отбора на поля блокировки по множеству значений используется поле ИсточникДанных и метод ИспользоватьИзИсточникаДанных. Источником данных могут быть объекты типов РезультатЗапроса, ТабличнаяЧасть, ТаблицаЗначений. В методе ИспользоватьИзИсточникаЗначений следует указывать поле блокируемого объекта и поле, из которого берутся значения для блокировки.

## **19) Правила внесения изменений в структуру регистров рабочей базы:**

Правила внесения изменений в структуру регистров рабочей базы:

1. Продумать все детали изменений регистра (какие будут структурные изменения, какие документы делают движения прихода, какие документы делают движения расхода по новым измерениям )

2. Внести структурные изменения в регистр. Следует учесть, что изменение структуры регситров не потребует для обеспечения правильности синтаксиса изменения текстов программных модулей, читающих данные из регистров. Тем не менее это может отразиться на скорости их исполнения, т.к. скорость доступа к первым измерениям выше, чем к последним.

3. Реализовать движения прихода по этому регистру(регистрам) и отладить написанный алгоритм проведения на примере одного документа, но для каждого вида документов приходования при различных вариантах исходных данных.

4. Реализовать движения расхода по этому регистру (регистрам) или движения сторно для РегистровРасчета, РегистровНакопления типа Обороты и Регистров Сведений и отладить написанный алгоритм проведения на примере одного документа, но для каждого вида документов расходования при различных вариантах исходных данных.

5. Убедиться в корректности работы всех программных механизмов , которые базируются на этом регистре (регистрах), то есть корректность работы отчетов, обусловленного проведения других документов.

Часть механизмов базируется на работе с набором записей нашего регистра, поэтому целесообразно посмотреть наличие обработчиков событий ПередЗаписью и ПриЗаписи в модуле набора записей регистра. Затем для просмотра всех остальных механизмов на панели инструментов Конфигурация следует вызвать кнопку Поиск во всех текстах. В полученном окне проставить условия поиска (например, Модули, Роли). Результат поиска будут отражены внизу в табло. Однократным нажатием на строку результата поиск можно перейти в объект, содержащий данные, по которым можно оценить корректность применения нового механизма.

6. Затем следует перепровести все документы, связанные с данным регистром. В случае невозможности проведения старых документов по новым законам придётся реализовать корректное использование нужного механизма в зависимости от даты документа.

## **20) Направления оптимизации алгоритма проведения:**

Направления оптимизации алгоритма проведения в условиях повышенных требований к быстродействию системы можно разделить на следующие варианты:

1.Минимизация количества обращений к базе.

2.Минимизация объема данных, обрабатываемых в процессе проведения ( минимизация количества таблиц БД, задействованных в алгоритме, минимизация размеров обрабатываемых таблиц, минимизация действий с таблицами БД).

3.Минимизация количества и времени блокировок.

4.Оптимизация алгоритмов обработки данных в части приоритета решений, ориентированных на повышение параллельности работы перед решениями, обеспечивающими дополнительный сервис пользователям.

Для этого целесообразно применять следующие решения:

1. Применять разделение обработки регистрации документов в учете следующим образом:

а. Действия, требующие моментальной обработки(в момент проведения документа);

б. Действия, которые можно отложить на выполнение позже ( в конце смены, только перед формированием регламентной отчетности и пр.)

2.Задача сокращения времени проведения расчета может быть решена с позиции структурного подхода. Например, часть рассчитанных в процессе проведения документа показателей можно сохранять в промежуточных регистрах. Это делается для того, чтобы после можно было обратиться к этим данным - не тратить время на повторный расчет при последующих обработках документов. Можно вводить в систему служебные показатели, которые не используются в отчетах, и повышают эффективность проведения документов.

3. Целесообразно сокращать время действия блокировок, накладываемых в процедуре проведения за счет максимального сдвига момента наложения блокировок как можно ближе к концу алгоритма проведения.

## **21)Оптимизация оборотных регистров. Агрегаты:**

Когда количество измерений в регистре >= 8 имеет смысл отключить использование данных измерений в итогах (флаг Использование в итогах) - это не только уменьшит таблицу итогов регистра, но и позволит системе быстрее формировать отчеты, которые не требуют аналитических разрезов по данным измерениям.

В ситуации создания систем с аналитическими отчетами различной детальности, при максимальном количестве разрезов свыше 8 необходимо разделять измерения регистров на обязательные ( используются почти везде и требуют быстрой обработки) и необязательные ( используются редко, не требуют быстрой обработки).

Использование таблицы итогов для регистров накопления имеет ряд ограничений:

1. Периодичность промежуточных итогов чаще всего равна месяцу.

2. Индексирование таблицы итогов строится в порядке следования измерений.

3. Таблица итогов изменяется в момент формирования движений, что приводит к её разрастанию.

В больших базах это приводит к значительному снижению быстродействия. Для повышения оперативности реализации разноплановых аналитических задач имеет смысл использовать таблицы Агрегатов. Эти таблицы являются таблицами итогов с задаваемым разработчиком составом разрезов и периодичностью хранения. При этом следует помнить, если используется механизм агрегатов, отключается механизм итогов, справедливо и обратно утверждение.

Эксплуатация агрегатов разделена на три этапа:

1.Создание агрегата : необходимо добавить агрегат, описать его состав и установить частоту его использования.

2.Перестроение сети агрегатов : Если частота использования агрегата автоматическая, тогда решение о необходимости заполнения и поддержания агрегата система принимает автоматически на основании накопленной статистики реального использования. Если за последний период агрегаты не повышали эффективность запросов, то до следующего перестроения агрегатов может быть принято решение о его временном выводе из эксплуатации. Вызов перестроения агрегатов можно осуществлять программно.

РегистрыНакопления.<ИмяРегистра>.ПерестроитьИспользованиеАгрегатов(). Эффективно прикрепить этот метод к регламентному заданию, которое правильнее выполнять раз в сутки или неделю в зависимости от частоты использования базы, в то время, когда в базе никто не работает. В случае большой базы эта операция может выполняться несколько десятков минут.

3.Обновление агрегатов. Пополнение таблиц агрегатов происходит не в момент формирования движений по регистру, а после запуска специального процесса обновления агрегатов. Это не влияет на корректность запросов. Когда система не находит данные в таблицах агрегатов, недостающие данные она получит из таблицы движений регистра. Вызов перестроения агрегатов можно осуществлять программно.

РегистрыНакопления.<ИмяРегистра>.ОбновитьАгрегаты().

# БУ

## 1)Ручная операция, уточнения по кор.счету:

Т.к. в разрабатываемой вами конфигурации присутствие документа РучнаяОперация является обязательным, это накладывает неявное, но обязательное требование - во всех запросах и отчетах, которые вы делаете для решения задачи, накладывать максимальное количество уточняющих условий на счета. К примеру, если в условии вашей задачи упомянуто, что документ приходная делает проводку Дт Товары - Кт Поставщики, а расходная - Дт ПрибылиУбытки Кт Товары то анализируя потом для какой-либо цели (зависит от задачи) всю эту ситуацию, удобно использовать таблицу Обороты, задавая условие счета равное счету Товары,а на выходе дебетовый оборот будет со счетом Поставщики, а кредитовый - ПрибылиУбытки. Но в условии виртуальной таблицы обязательно нужно указать условие, приведенное ниже, и тогда РучнаяОперация, какая бы корреспонденция счетов там ни была введена, не сможет нарушить работу вашего запроса:   
  
| КорСчет В (Значение(ПланСчетов.Управленческий.Поставщики), Значение(ПланСчетов.Управленческий.ПрибылиУбытки))

## 2) Флаг "Балансовый":

Флаг "Балансовый" означает, что для такого измерения в наборе записей регистра с поддержкой корреспонденций будет создано одно поле, и любая проводка будет отмечаться как относящаяся (целиком - и Дт и Кт) к данному измерению (напр. Организации).

## 3) Реализация аналитического учёта :

Аналитический учёт может быть реализован в платформе в одном из трёх вариантов :  
  
1.Сквозная аналитика, при которой один и тот же аналитический разрез используется на всех или почти на всех счетах (напр. Организация, Центр учета или КБК). В данном случае аналитический признак (например, подразделение) привязывается не к проводке в целом, как, например, при консолидированном учете, а только к одной ее стороне (Дт или Кт), вполне допустимо в рамках одной проводки выполнить перемещение средств или обязательств с одного подразделения на другое.  
В ПланеСчетов создали признак учета (УчетПоПодразделения) для возможности ведения учета по данной аналитике на необходимых счетах.Добавили небалансовое измерение Подразделение, для него установили свойство ПризнакУчета в значение УчетПоПодразделениям. После этого в реальной таблице регистра бухгалтерии добавлены поля ПодразделениеДт и ПодразделениеКт для отражения Подразделения Дт и Подразделения Кт в проводке(Для некоторых счетов эти поля будут не заполнены). В таблице ОстаткиОбороты РБ появилась колонка "Подразделение" , в таблице ОборотыДтКт появились поля ПодразделениеДт и ПодразделениеКт.   
  
2.Обычная аналитика, при которой на разных счетах должна присутствовать различная аналитика в разном количестве параллельных срезов. В данном случае необходимо:  
1. задать все возможные виды субконто (ПВХ ВидыСубконто);  
2. указать максимально возможное количество видов субконто  
3. задать на счетах субконто в требуемом постановкой задачи порядке.  
  
3.Опциональная аналитика, которая является разновидностью обычной, но может добавляться или убираться со счетов пользователем. В данном случае необходимо задействовать механизм функциональных опций и констант.

## 4)Рекомендации по определению субконто:

Лучше определять субконто на счете в порядке убывания значений в них. Например, в справочнике Номенклатура - 100000 позиций, в справочнике Контрагенты - 10 позиций. На счете товары первым добавлено субконто "Номенклатура", вторым - "Контрагенты". Целесообразно устанавливать один и тот же вид субконто на одну и ту же позицию на разных счетах. Аналитическая отчетность по всем сче6там, имеющим одинаковую аналитику формируется быстрее при такой организации .

5) БУ: При добавлении одной проводки в регистр бухгалтерии обновляются следующие таблицы:  
1.Таблица движений: добавляется одна строка для этой проводки;  
2.Значения субконто: от нуля до 4 строк , по одной строке на каждое субконто обоих счетов;  
3.Остатки и обороты по счетам: обновляются две строки - для счета Дт и для счета Кт.  
4.Остатки и обороты по счетам и субконто: обновляется одна строка для каждого счета проводки в одной из двух таблиц в зависимости от того, сколько видов субконто прикреплено к счету. Если счета синтетические - таблицы итогов по субконто не обновляются;  
5. Обороты между счетами: обновляется одна строка итогов по корреспонденции счетов.

## 5) Индексирование таблиц регистра бухгалтерии:

Индексы используются не только для быстрого получения итога в отчетах, но и для поиска строки итогов при записи движений в регистр. Максимальное количество индексируемых полей - 16. По индексу система быстро находит строку, которую необходимо заблокировать для внешних изменений и изменить. Когда в индекс помещаются не все поля, система получает диапазон строк и в дальнейшем перебирает его в поисках нужной строки - это снижает скорость выполнения кода.  
  
Таким образом включать примитивные типы данных в состав субконто можно только в случаях, когда это - единственный вариант реализации. Они увеличивают не только количество индексируемых полей , но и количество индексов. Кроме того, при проектировании регистра бухгалтерии и аналитики ( субконто ) необходимо, чтобы количество индексируемых полей не превышало 16.  
  
В физических таблицах регистра бухгалтерии индексируются Период, счет, Измерения и Субконто. Для хранения одного значения составного типа данных программа расходует 3 поля в базе данных: (1)имя объекта, (2) имя таблицы, (3)ссылка на запись) , при условии, что в состав входят значения ссылочного типа. Для каждого значения примитивного типа данных в базе данных будет задействовано еще одно поле. Три обязательно индексируемые поля это - Период, Счет и Разделитель(когда включен режим разделения итогов).

## 6) Анализ результата запроса, когда неизвестно на каком месте находится какое субконто:

Анализ результата запроса, когда неизвестно на каком месте находится какое субконто происходит следующим образом.   
1.Необходимо задать параметр ВидСубконто в виртуальной таблице РегистраБухгалтерии. В него можно передать ссылки или массив ссылок на вид(виды) субконто.  
2.Порядок следования элементов в массиве задает порядок группировок в запросе.  
  
Отбор по видам субконто не нужен, когда делаем запрос по одном счету, на котором субконто задано в конфигураторе.

## 7)Получение развернутого сальдо из виртуальных таблиц:

Для получения развернутого сальдо из виртуальных таблиц регистров бухгалтерии следует использовать поля РазвернутыйОстатокДт и РазвернутыйОстатокКт . Чтобы В СКД получать развернутые остатки по аналитике, когда их нет в составе выбранных полей следует воспользоваться свойством "Роль" доступных полей компоновки. Пример:  
В роли измерений (Субконто1, Подразделение), которые должны остаться в выборке запроса вне зависимости от выбора пользователя в диалоге настроек выбранных полей отчета, следует активировать флаг "обязательное". Таким образом остатки будут развернутыми в любом случае.

## 8) Количественный учет

Этот вид учета имеет смысл не на всех счетах, поэтому требует создания признака учета Количественный, чтобы отметить счета, на которых будет вестись учет ценностей в натуральном выражении. Учет ведется в отдельном ресурсе регистра бухгалтерии, при этом он допускает операцию комплектации номенклатуры.Данный ресурс является небалансовым и создает в таблице движений регистра бухгалтерии с поддержкой корреспонденции два поля : "КоличествоДт" и "КоичествоКт".Баланса двойной записи по этому ресурсу нет. Например, можно списать с кредита счета n единиц, а в дебет счета получить 1 единицу.(Комплектация номенклатуры).  
  
Во все таблицы остатков и оборотов для хранения остатков и оборотов в количественном выражнении (ОстаткиОбороты по счетам и ОстаткиОбороты по счетам и субконто) добавлены три новых поля: Остаток. ОборотДт и ОборотКт. в таблице ОборотыМеждуСчетами добавлены два поля: отдельно для хранения оборотовДт и оборотовКт. Новые поля отличаются от созданных ранее приставкой Количество. В таблице Обороты появилась возможность анализировать кор.обороты, в таблице ОборотыДтКт количественных оборотов - 2, что соответствует ее источнику физической таблице итогов ОборотыМеждуСчетами.

## 9) Валютный учет:

Валютный учет:  
  
Данный вид учета используется для учета активов, выраженных в иностранной валюте, которая учитывается чаще всего в кассе и на банковских счетах.Валютный учет подразумевает хранение в итогах на отдельных (не на всех) счетах остатков им оборотов не только в валюте учета, но и в иностранных валютах, причем в разрезе иностранных валют.  
  
Для реализации задач валютного учета чаще всего требуется :  
  
1. Небалансовый ресурс для хранения еще одной характеристики регистра  
2. Небалансовое измерение для дополнительного разреза счетов по валютам  
3. Признак учета счета, чтобы отключать валютный учет на тех счетах, где он не нужен  
  
Дополнительно потребуются:  
  
4. Периодический регистр сведений КурсВалют с периодичностью в день (в нем хранится значение курса и кратности (для величин курс которых меньше курса валюты учета)). Если нужен доступ к курсам валют из справочника, необходимо измерение Валюта сделать Ведущим, и нужна форма элемента справочника "Валюты", в командном интерфейсе которой должен быть доступ к регистру сведений  
  
В регистре бухгалтерии следует добавить небалансовый ресурс ВалютнаяСумма и небалансовое измерение Валюта. Небалансовые свойства регистра добавят в набор записей свою валюту и свою валютную сумму для Дт и Кт проводки. Такая реализация правильная, потому что:  
не все счета будут балансовые, поэтому баланс двойной записи по этому ресурсу и измерению не нужен. Возможны операции, когда задолженности в валюте будут гаситься средствами в другой валюте и на другую валютную сумму.  
  
Для задач валютного учета полезно вынести функции пересчета сумму по курсу во внешний общий модуль, для которого установлен активным флаг Сервер.(ВызовСервера неактивен, т.к. данный метод вызываем из модуля документа, который выполняется на Сервере, а вызовов данного метода с клиента не будет).  
  
После добавления небалансового измерения и небалансового ресурса во всех таблицах итогов добавится новый разрез по Валюте и колонки для хранения остатков и оборотов в валюте. В таблице ОборотыМеждуСчетами добавляются два поля: отдельно для хранения дебетового и кредитового оборота, а также разрезы ВалютаДт и ВалютаКт. В виртуальных таблицах появилась возможность отбора группировки данных по валюте и анализа остатков и оборотов ресурса "Валютная сумма".  
  
При построении отчетов вида ОСВ по валютам на СКД в задачах валютного учета следует не забыть указать поле расчета Валюта на вкладке Ресурсы, чтобы не получить сумму по всем учитываемым валютам  
  
Курсовые разницы: Если изменяется курс валюты возникают курсовые разницы, которые должны быть рассчитаны, а затем по ним формируются проводки. Курсовые разницы возникают при сравнении остатков средств (обязательств), выраженных в иностранных валютах и их эквивалентов в валюте учета. При этом пересчет валютных остатков в рублевые осуществляется по актуальному курсу валюты на день пересчета.  
Формула для расчета курсовой разницы как правило имеет вид:  
  
КурсоваяРазница = ВалютнаяСуммаОстаток \* Курс / Кратность - СуммаОстаток  
  
Источниками данных для расчета курсовых разниц служат таблица Остатки РегистраБухгалтерии и таблица срез последних регистра сведений Курс. Соединение на практике используют левое, причем слева - остатки. Внутреннее соединение следует использовать для того, чтобы исключить итоги по валюте с неустановленными курсами. Если объем данных большой, правильнее работать с временными таблицами , использовать предварительно сохраненные результаты их выполнения.

## 10) Документ РучнаяОперация:

При автоматизации ОУ документ, который позволяет сделать любую произвольную запись в регистр - редкое явление. Главная причина - регистр накопления почти не используется пользователями. Они работают с документами и отчетами. В БУ пользователю необходимо формировать произвольные движения в регистре бухгалтерии, кроме того, все бухгалтерские операции не могут быть реализованы "от документа", поскольку это - неэффективно. По этой причине необходим документ РучнаяОперация или КорректировкаДвижений, который позволяет пользователю изменять набор записей регистра непосредственно в форме документа и вводить в регистр любые произвольные движения.  
  
Проведение данного документа запрещено, он является регистратором для РегистраБухгалтерии. В случае ведения многофирменного учета в одной базе в РучнуюОперацию добавляют РеквизитОрганизация для   
правильного ведения учета.  
  
В обработчике события записи документа ПередЗаписью необходимо прописать код перебора проводок из табличной части формы для того, чтобы заполнить в движениях Период и Организацию.  
  
Кроме того необходимо обработать ситуацию с копированием документа.  
В Обработчике События ПриКопировании следует прописать код:  
  
"...  
Проводки = ОбъектКопирования.Движения.Проводки.Прочитать();  
  
Для каждого ИсходнаяЗапись Из Проводки Цикл  
  
Проводка = Движения.Проводки.Добавить();  
ЗаполнитьЗначенияСвойств(Проводкка,ИсхЗапись);   
  
КонецЦикла;  
  
..."  
  
В диалог формы документа следует перенести набор записей регистра бухгалтерии "Проводки"  
  
Активностью поводок следует управлять из модуля документа РучнаяОперация .Когда пользователь помечает этот документ на удаление активность движений, подчиненных данному документу отключается. При снятии пометки удаления активность можно будет снова включить.В процедуре ПередЗаписью модуля объекта РучнаяОперация необходимо прописать код:  
  
"...   
Если НЕ ЭтоНовый() И Ссылка.ПометкаУдаления <> ПометкаУдаления Тогда  
  
Движения.Проводки.Записывать = Истина;  
Движения.Проводки.Прочитать();  
Движения.Проводки.УстановитьАктивность(НЕ ПометкаУдаления);   
  
КонецЕсли;  
..."  
  
При изменении пометки удаления документа в свойство документа ПометкаУдаления записывается новое значение, после чего документ записывается в базу данных. Это событие и перехвачено в обработке.

## 11) Объект НаборЗаписей регистра бухгалтерии

Объект НаборЗаписей регистра бухгалтерии на практике можно использовать для заполнения новых свойств регистра за прошедшие годы.   
  
Пример:  
  
Пользователь хочет увидеть прибыли и убытки (Дт и Кт счета Капитал) в разрезе номенклатуры для анализа выгодности торговли конкретным видом товара. Для этого следует на счете Капитал разместить оборотное субконто Номенклатура. Субконто вида Остатки создаст много проблем за предыдущие периоды, а субконто вида Обороты - нет. Для решения поставленной задачи предварительно нужно изучить все учетные ситуации: какие могут быть корреспонденции, откуда брать нужную информацию, и разработать обработку,которая без проведения заполнит новое субконто. При обработке больших массивов записей необходимо перед обработкой отключать использование итогов - это ускорит запись движений в регистр.Разумеется, по окончании записи, использование итогов нужно будет снова включить.

## 12) МодульНабораЗаписей регистра бухгалтерии

Модуль набора записей часто используется для заполнения свойства регистра, значения которых получаются расчётным путём и одинаковы для всех документов. Примером такого учёта может быть трёхвалютный учёт, при котором весь учёт ведётся параллельно в двух валютах: в основной валюте учета и в еще в одной, причем по историческому курсу. Это означает, что при записи каждой проводки надо параллельно записать пересчитанную по курсу рубля на дату проводки сумму этой проводки еще в евро. Для решения такой задачи потребуется создать новый ресурс регистра бухгалтерии СуммаХолдинга. Этот ресурс - балансовый, аналогичный по свойствам ресурсу Сумма,Разница только в том, что этот ресурс заполняется автоматически. Потребуются две константы ВалютаУчета и ВалютаХолдинга. В общем модуле следует описать код, рассчитывающий отношения курсов валют (кроссКурс).  
  
В модуле набора записей необходимо прописать код в обработчике события ПередЗаписью:  
"...   
Проводки =ЭтотОбъект;  
Если Проводки.Количество() = 0 Тогда  
 Возврат;  
КонецЕсли;  
  
Период = Проводки[0].Период;  
ВалютаУчета = Константы.ВалютаУчета.Получить();  
ВалютаХолдинга = Константы.ВалютаХолдинга.Получить();  
КроссКурс = ОбщийМодуль.ПолучитьКроссКурсВалют(ВалютаУчета, ВалютаХолдинга,Период);  
  
Для Каждого Проводка Из Проводки Цикл  
  
Если Проводка.СуммаХолдинга <> 0 Тогда  
 Продолжить;  
КонецЕсли;  
  
Проводка.СуммаХолдинга = Проводка.Сумма \*КроссКурс;  
  
КонецЦикла;  
..."

## 13) Признаки учета субконто:

Основная цель ПУС - управлять хранением итогов ресурсов регистра в разрезе субконто. ПУС позволяет в рамках замкнутой системы показателей БУ решать такие задачи как отказ от остатков с хранением только оборотов или обеспечение для разных ресурсов разного количества разрезов. ПУС позволяет отказаться от хранения ненужных для субконто итогов , сохранив итоги по счету в целом.  
Коротко, ПУС должен быть связан с ресурсом, итогами которого он управляет.  
  
Пример:  
  
Отчет ДвижениеДенежныхСредств  
  
Остатки на начало и конец нужно хранить без постатейной аналитики (Оплата от поставщика всегда будет величиной положительной, неинформативно накапливать такие сведения),  
а обороты за период целесообразно получать с разбиением на статьи. В ОУ подобную задачу можно реализовать на 2 регистрах накопления: регистр остатков и регистр оборотов(Статьи движения денежных средств). В БУ один из вариантов реализации данной задачи имеет следующий вид: 1.Создать вид субконто ДвиженияДенежныхСредств. 2.Добавить субконто на счет Касса. 3. Активировать для субконто ПУС ТолькоОбороты  
  
На итоги Регистра Бухгалтерии этот флаг повлияет следующим образом.  
Остатки и обороты по счетам хранятся в нескольких таблицах: в одной - остатки и обороты с разрезом только по счетам и измерениям, в нескольких таблицах (столько сколько задействовано субконто) хранятся итоги в разрезе субконто. После добавления ПУС по счету Касса по - прежнему хранятся остатки и обороты. В таблице, которая хранит итоги по первому субконто в поле Остаток записан 0. Если в текущем периоде не будет оборотов, то строка в этой таблице итогов создана не будет( при хранении остатков строка будет создана на каждый месяц, чтобы хранить остаток с прошлого месяца, даже, когда нет оборотов).При обращении к остаткам в виртуальных таблицах, если установить отбор и задать группировку по субконто и платформа обратится по субконто к таблице итогов по субконто, результат запроса будет содержать нули.  
  
Таким образом таблица итогов по счетам и первому субконто после ввода ПУС остатки не хранит.  
  
Чаще всего ПУС применяется для реализации количественного учета по складам. Т.е. при списании товара со склада необходимо знать остатки по складу, а при расчете себестоимости склад надо игнорировать, и рассчитывать себестоимость списания по предприятию в целом.  
  
Пример:   
  
В ОУ подобная задача решается на двух регистрах накопления: в одном хранится количество и стоимость номенклатуры (КиС), во втором - количество номенклатуры на складах(К).Расчет себестоимости происходит по данным первого регистра(КиС), контроль остатков по данным регистра (К). В БУ для решения данной задачи целесообразно увеличить количество строк регистра бухгалтерии, которые будут отдельно хранить количественные и стоимостные показатели.   
  
Иными словами необходимо добавить такие ПУС, какие виды учета требуется отключить. поскольку нужно отключить суммовой учет на складе, в план счетов был добавлен ПУС Суммовой .Этот признак установлен для ресурса Сумма, т.е. этот признак учета включен для всех субконто по умолчанию. На нужных счетах благодаря вводу ПУС суммовой учет по сумме можно отключить.  
  
После таких манипуляций при выполнении запроса к виртуальным таблицам остатков и оборотов, если по складу не установлен отбор/ группировка, тогда можно получить остатки по сумме и количеству, иначе - если отбор есть- только остатки по количеству. Это связано с тем, что платформа берет значения остатков из разных строк.(Со вводом ПУС строк прибавилось и таблица ОстаткиОбороты увеличилась).

# ОУ:

## 1) Корректировка записей регистров:

Данный документ необходим, когда логику установки или изменения значения показателей в регистре не предоставляется возможным, однако, это необходимо пользователю.

Документ не делает программных движений, при этом движения записываются непосредственно в регистр. Этот документ служит для формирования движений по регистру накопления вручную. Таблицу движений документа следует вынести на форму документа, а также вывести на форме документа ссылку на форму списка регистра накопления, по которому документ формирует движения.

Для обеспечения корректности использования системы способ "интерактивного формирования движений" целесообразно ограничить действия пользователя настолько, чтобы пользователь не смог нарушить корректность формирования движений (например, не мог ввести с привязкой к некоему документу записи с указанием абсолютно разных периодов, что может привести к проблемам с достоверностью ведения учета) .

Вопрос соответствия периода сформированных движений периоду регистрации документа решается следующим образом. В обработчике события записи ПередЗаписью модуля набора записей регистра следует прописать следующий код:

"...

Документ = ЭтотОбъект.Отбор.Регистратор.Значение;

Для каждого ТекЗапись Из ЭтотОбъект Цикл

ТекЗапись.Период = Документ.Дата;

КонецЦикла;

..."

Система перед записью набора записей регистра определяет дату документа - регистратора и присваивает это значение полю период каждого элемента (строке) из коллекции набора записей.

Следует запретить запись данного документа с датой, превосходящей текущую, поскольку подобные записи будут нарушать концепцию контроля отрицательных остатков.(Движение товара, запланированное на будущее, исказит остатки, при этом движения товара фактически нет). для этих целей в модуле объекта документа КорректировкаЗаписиРегистров в обработчике события записи ПередЗаписью следует прописать следующий код:

"...

Если НачалоДня( Дата ) > НачалоДня ( ПолучитьОперативнуюОтметкуВремени() ) Тогда

Отказ = Истина;

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Нельзя вводить документы этого типа будущей датой " ;

Сообщение.Сообщить();

КонецЕсли;

..."

## 2) Оптимизированный алгоритм проведения документа

Оптимизированный алгоритм проведения документа содержит минимальное количество обращений к БД (чтение данных, запись сформированных данных). Процесс разработки алгоритма проведения документа должен быть эффективным и содержать минимум критических точек(потенциальных ошибок). Для этого следует придерживаться следующих правил:

1.Разработку алгоритма проведения следует начинать с конструктора движений, поскольку быстрее и правильнее движения формируются с помощью этого инструмента проектирования.

2.В построенной конструктором процедуре ОбработкаПроведения разметить комментариями тело обработчика проведения. В комментариях изложить недостающие действия (по формированию движений).

3. Если по комментариям видно, что для этих действий достаточно данных только самого документа, значит, имеем дело с безусловным проведением. То есть для реализации алгоритма будет допустимо использовать только объектную модель получения данных - в контексте обработчика находится весь объект документа ( все необходимые для формирования движений данные). На этом алгоритм можно завершить. Если проведение документа - обусловленное, тогда необходимо получать недостающие данные запросом, схему которого имеет смысл разработать по п.31. Рекомендуется предусмотреть крайние ситуации. Например, когда продаваемого товара нет на складе или когда один и тот же товар несколько раз зафиксирован в табличной части документа.

4.Следует отработать текст запроса в Консоли запросов, собирая запрос от простого к сложному, с пошаговым выполнением запроса. Это будет эффективнее отладки сложного запроса.

5. Отлаженный текст запроса необходимо, по возможности. оптимизировать под конкретную задачу, после чего текст запроса следует перенести в обработчик проведения.

6. Затем следует реализовать корректную обработку результата запроса со всеми возможными проверками.

## 3) Правило упрощения реализации запроса по нескольким источникам

Для облегчения реализации запроса по нескольким источникам допускается руководствоваться правилом:

Если на схеме запроса видно, что один из источников может добавлять новые группировочные сущности ( например, новые номенклатурные позиции), то лучше применять ОБЪЕДИНЕНИЕ, если один из источников добавляет информацию только в новые колонки, то в этом случае правильнее использовать СОЕДИНЕНИЕ.

## 4) Виртуальные таблицы

1.РегистрНакопления.<ИмяРегистра>.Остатки(<Период>, <Условие>)

При получении данных из виртуальной таблицы Остатки, когда период не указан, данные получают за интервал по самую последнюю дату включительно. Когда параметр Период задан, тогда данные получают на начало указанного значения временной шкалы. (Следует использовать МоментВремени() вместо данных поля Дата). Поле Условие используется для ограничения состава записей, по которым будут выбраны итоги. Условие применяется исходным записям, которые содержатся в регистре, поэтому наложение условия в параметрах виртуальной таблицы повышает быстродействие запроса. В регистры накопления почти всегда заносят движения, у которых флаг Активность установлен в значение Истина. Именно они влияют на итоги. Неактивные движения записывают в исключительных случаях, например, когда событие необходимо отразить в базе, но оно не должно повлиять на итоги.

2.РегистрНакопления.<ИмяРегистра>.Обороты(<НачалоПериода>, <КонецПериода>, <Периодичность>,<Условие>)

Отбор по оси времени происходит с включением граничных точек, однако, параметр КонецПериода следует задавать с использованием методов получения конечной даты периода ( например. КонецДня(Дата)). В противном случае КонецПериода будет содержать значение даты по умолчанию, т.е. на начало указанной даты.

Когда данные в запросе надо получить не за интервал, а с разбиением, то используется параметр <Периодичность>. С его помощью можно задавать указание дополнительного разворота итогов по периодичности. Данный параметр принимает значения : Период - обороты не разворачиваются;

Регистратор - обороты разворачиваются по периоду регистрации документа - регистратора;

День, Неделя, Месяц, Квартал, Год - по выбранному периоду. В случае варианта Авто за счет выбора выходных полей можно создавать разворот внутри разворотов, то есть развернуть показатели по годам, при этом каждый год развернуть по кварталам , кварталы - по месяцам и далее.

Поле Период результата запроса существует только в том случае, если параметр "Периодичность" при формировании таблицы получил не пустой значение. Поле Регистратор также существует, если параметр Периодичность был установлен при формировании запроса в значение "Регистратор" или "Авто".

3.РегистрНакопления.<ИмяРегистра>.ОстаткиИОбороты(<НачалоПериода>, <КонецПериода>, <Периодичность>,<МетодДополнения>, <Условие>)

Данная таблица есть комбинация виртуальных таблиц Остатки и Обороты. По данной таблице можно получать данные о начальных остатках, оборотах и конечных остатках из одной таблицы, что снижает количество блокируемых данных в транзакциях и повышает достоверность и эффективность одновременно, поскольку нет необходимости синхронизировать данные нескольких регистров - все данные хранятся в одном регистре.

Параметр МетодДополнения определяет необходимость показа данных периода (- ов) между НачалоПериода и КонецПериода, в которых не было движений.

## 5) Реализация партионного учета

В документе поступления обязательно должно присутствовать измерение партия (Документ.Ссылка), которое принимает значение Ссылка на проводимый документ. В документе списания обработчик проведения содержит зачастую типичный порядок действий:

1.Для каждой из строк табличной части документа списания необходимо убедиться , что товара для списания достаточно на момент проведения.

2.Затем следует выбрать остатки всех партий каждого товара, партии упорядочить в соответствии с методом списания (FiFo/ LiFo) по полю МоментВремени() документа - партии.

3. После чего следует, поочередно перебирая остатки партий, принять решение списывать партию полностью или только часть партии. Необходимо не списать лишнего, т.е. контролировать количество к списанию после списывания товара по партии.

4.В запросе потребуются сведения об остатках с детализацией по партиям, это значит, что необходимы итоги по количеству и сумме номенклатуры для контроля остатков. При этом количество списания не суммируется при получении итогов а выбирается агрегатной функцией (МИНИМУМ, МАКСИМУМ или СРЕДНЕЕ)

5. Результат запроса нужно обойти сначала по группировкам, а затем по детальным записям.

6. При обходе по детальным записям, если количество списания <= КоличествоОстаток по номенклатуре, количество и сумму партии списываем полностью. В противном случае, если КоличествоОстаток отлично от 0, списываем часть партии.

## 6)Организация планирования процесса оказания постпродажных услуг. Работа с регистром сведений.

Рассмотрим пример.

Необходимо реализовать учетную схему, в которой отражается планирование оказания услуг, оказание услуг и контролируется какие услуги проданы, но не запланированы к исполнению, какие услуги запланированы, но не выполнены, когда были выполнены запланированные услуги.

Эффективнее всего хранить такую информацию в регистрах сведений, поскольку он оперирует показателями состояния.

Поскольку стоит задача хранить не только актуальные состояния выполнения услуг, но и иметь возможность оценить хронологию состояний, регистр сведений будет периодическим. Кроме того, это позволяет получать на любой момент времени срез актуальных значений.

Для корректного проектирования регистра сведений важно понимать, что в любом регистре должны существовать записи только абсолютно уникальные по набору значений ключевых полей. К этим полям относятся :

1. измерения,

2. поле Период, когда Регистр Сведений периодический

3. поле Регистратор, когда Регистр Сведений подчинен документу - регистратору и периодичность регистра сведений установлена в значение " по позиции регистратора".

Если в момент среза для сущности необходимо получить много значений одновременно, сущность нужно оформить в виде измерений. Если сущность для конкретного набора измерений существует в единственном экземпляре - это ресурс.

Этапы работы с спроектированным регистром сведений ПланированиеОказанияУслуг можно выделить следующим образом:

Измерения: СервисМенеджер,Услуга,Контрагент,ДокументОснование

Ресурсы: ПланируемаяДатаВыполнения, РеальнаяДатаВыполнения

Документы - регистраторы: ПродажаТоваров, ПланированиеОказанияУслуг, ОказаниеУслуг

1.Возникновение потребности в планировании выполнения услуги.

2.Планирование выполнения услуги : заполнение и проведение документа.

3. Оказание услуги: заполнение и проведение документа.

4. Формирование отчетности

# СКД

Чтобы отчет не рассчитывал лишние итоги по ненужным измерениям следует на вкладке ресурсы указывать, по какому полю нужно вести расчет.

# ПР

## 1) Период регистрации:

Период регистрации - период (месяц) за который результаты месячного расчета зарплаты предполагается отразить в бухучете . Его также называют учетный период.

Месячный заработок для сотрудника, работающего повременно рассчитывается по формуле

СуммаНачисления = Оклад \* (ФактическиОтработанноеВремя/ЧислоРабочихДнейВМесяце)

Если сотрудник помимо оклада получает премию, заработок будет исчисляться следующим образом :

СуммаНачислений = СуммаОклада + СуммаОклада \*( ПроцентПремии /100);

Формулы расчета Оклада и Премии отличаются, а в общем случае таких составляющих суммы начислений может быть несколько. В зарплатных решениях такие составляющие суммы начисления, которые рассчитываются сотрудникам при начислении зарплаты и каждый из которых имеет собственную расчетную формулу, называются видами расчета.

Сумма, складывающаяся из результатов других видов расчета, по отношению к данному виду расчета (Премия) называется суммой базовых начислений или просто: расчетной базой, базовой суммой, суммой базы или базой. Алгоритм называется - расчет по базе. Прежде чем приступить к расчету, необходимо иметь результаты расчета базы всех видов расчета, которые мы хотим рассчитать. Кроме состава базовых видов расчета необходимо знать порядок, в котором осуществляется их расчет.

Иными словами нужно обеспечить:

1. Хранение состава базовых видов расчета.

2. Необходимый порядок расчета : " снизу вверх по дереву базовой зависимости.

Начисления - виды расчета, которые складывается с общей суммой рассчитанной зарплаты, а удержания - те, которые из нее вычитаются.

Для некоторых видов расчета, которые предполагают протяженность по времени предполагается планируемый период действия, известный заранее. В течение него этот вид расчета будет действовать. Этот планируемый период, известный заранее, называется периодом действия ВидаРасчета.

Проще говоря - ПериодДействия - это тот месяц, за работу в котором сотруднику производится расчет. Следует отметить, что ПериодРегистрации с ПериодомДействия может не совпадать. Например, начисление сотруднику можно рассчитать в мае (период действия), а провести - в июне (период регистрации) или другом месяце. Эти периоды имеют разное смысловое содержание и могут как совпадать, так и не совпадать.

Начисления по видам расчета, зависящим от отработанного времени по расчетной формуле, могут зависеть друг от друга ( Оклад и Командировка). Эта зависимость называется зависимостью по периоду действия, а виды расчета, прерывающие действия связанных с ними видов расчета являются по отношению к ним вытесняющими видами расчета.

Совокупность несмежных интервалов, на которые разбивается изначальный непрерывный период действия вида расчета, называется фактическим периодом действия периода расчета.

## 2) Состав расчетных механизмов платформы:

Состав расчетных механизмов платформы следующий:

1. Механизм получения суммы базовых начислений;

2. Механизм получения суммы рабочего времени по фактическому периоду действия, а также по периоду действия.

Данные, необходимые для расчета, разделяют на :

1. Исходные данные - они известны до начала расчета зарплаты(например, размер Оклада);

2.Необходимые данные - их предстоит получить в течение самого сеанса расчета зарплаты (например, ФактическийПериодДействия)

Виды расчета хранятся в объектах типа ПланВидовРасчета (ПВР) . Настройка данных объектов происходит определенным образом:

1. В первую очередь необходимо хранить информацию о расчетной формуле, по которой будет рассчитываться каждый вид расчета. Для этого необходимо создать реквизит ВидРасчета расчета - идентификатор расчетной формулы, под которым он используется в системе. При расчете суммы начисления необходимо анализировать значение этого реквизита, чтобы корректно рассчитывать Сумму.

2. На видах расчета может существовать базовая зависимость, в этом случае, при сложной системе зависимостей необходимо задавать очередность расчета видов расчета во времени, чтобы к моменту начала расчета такого вида, в расчетную формулу которого входит сумма базы, все виды расчета, результаты которых должны входить в сумму базы, были уже рассчитаны. Для этих целей необходимо создавать реквизит Приоритет(КатегорияРасчета), значение которого имеет тип Число. Виды Расчета с наименьшим приоритетом будут рассчитаны первыми.

## 3) Настройка расчетных свойств плана видов расчета:

Специальные расчетные свойства объекта ПВР настраиваются на закладке "Расчет". Их значения используются расчетными механизмами платформы.

Свойство "Использует период действия"

Свойство указывает расчетным механизмам, будут ли в ПВР храниться ВР, которые по расчетной формуле зависят от отработанного времени. Если будут, тогда для таких ВР может быть задана зависимость по периоду действия от других ВР. Т.е. для каждого такого ВР может быть задан состав вытесняющих ВР, которые будут разбивать период действия данного ВР.

Когда этот флаг установлен в значение Истина, система создает для ПВР стандартную табличную часть вытесняющих ВР, в которую можно вводить для каждого ВР состав вытесняющих ВР. Когда флаг неактивен - вытесняющих ВР не будет.

Свойство "Зависимость от базы"

Когда свойство установлено в значение "Не зависит", в данном ПВР нельзя будет хранить ВР, формула расчета которых зависит от суммы базы.

Понятие базового периода

Рассмотрим пример:

Если сотруднику нужно рассчитать начисление Надбавка , имеющее зависимость от базы начислений за январь 2016 года, то будет абсолютно неверно суммировать базовые начисления за все время его работы на предприятии, а нужно получать их за тот интервал, за который необходимо получить сумму начислений. Если ВР Надбавка рассчитывается помесячно, то суммировать результаты базовых начислений, от которых зависит Надбавка нужно за месяц. Интервал, который задается для выборки и суммирования результатов базовых ВР называется базовым периодом.

Таким образом в расчетных задачах выделяют 4 вида времени:

1 . Период регистрации

2. Период действия

3. Фактический период действия

4. Базовый период

## 4) Выбор критерия попадания в базовый период:

Выбор критерия попадания в базовый период - установка значения свойства "Зависимость от базы"

Таким образом для функционирования расчетного механизма получения базовой суммы начисления необходимо задавать базовый период.

Однако возникает неясная ситуация - расчетный механизм обнаруживает сумму результата расчета по базовому начислению, механизму перед началом расчета заданы границы интервала базового периода, но неясно как расчетный механизм определит относится результат расчета к базовому периоду или нет.

На данном этапе расчетному механизму известны период действия и период регистрации начисления. Таким образом, механизму получения суммы базы можно задать один из вариантов критерия суммирования результата базового вида расчета:

а.Попадание периода действия этого результата расчета в базовый период : Зависимость по ПД

б.Попадание периода регистрации этого результата расчета в базовый период : Зависимость по ПР

Критерий выбора свойства заключается в следующем:

Для ВР, которые являются начислениями логично воспользоваться критерием попадания периода действия в базовый период.

Например, если сотрудник получает оклад за январь 2016 г., то и для расчета январской надбавки используем значение январского оклада, т.е. критерий попадания в базовый период - ЗависитПоПериодуДействия. Для всех ВР, которые используют расчетную формулу с суммой базы за базовый период, расчетный механизм суммирует результаты базовых ВР за этот период.

Критерий попадания в базовый период ЗависитПоПериодуРегистрации рекомендуется использовать для расчета удержаний. Правильность расчета удержаний контролируется внешними организациями, а они для проверки используют данные бух.отчетности предприятия, в бух. отчетности используется только период регистрации. Если использовать для расчета таких ВР критерий попадания в базовый период действия, то внешние организации не смогут проверить правильность расчета удержаний, поскольку им известен только период регистрации (по данным БУ), а период действия неизвестен. Поэтому для расчета удержаний используем критерий попадания в базовый период периода регистрации.

Для ПВР, в которых хранятся виды начислений, свойство " Зависимость от базы" устанавливается в значение ЗависимостьПоПериодуДействия.

Для ПВР, в которых хранятся виды удержаний, свойство "Зависимость от базы" устанавливается в значение ЗависимостьПоПериодуРегистрации.

## 5) Вытесняющие виды расчета:

Вытесняющие виды расчета должны находится в одном ПВР с вытесняемыми ВР, т.к. установить вытесняющие ВР можно только из этого же плана расчета. Базовые ВР, в отличие от вытесняющих, могут располагаться как в одном так и в разных ПВР относительно зависимого ВР.

## 6) Регистры расчета:

Регистры расчета можно представить в виде двух компонентов:

1.Хранилище первичной информации;

2.Вычислитель расчетных показателей, который рассчитывает следующие показатели:

а. кол-во рабочего времени по плану расчетного периода (ПД);

б. кол- во фактически отработанного времени (ФПД);

в. сумма базовых начислений.

Расчетные решения работают по следующему алгоритму:

1.В регистр расчета поступают исходные данные, которые нужны для получения необходимых данных: период действия, состав базовых ВР и т.д.

2. Выдаются запросы к виртуальным таблицам РР, результаты выполнения которых будут содержать необходимые данные:

а. Кол - во рабочего времени (плановое и фактическое)

б. Сумма базовых начислений

3. Полученные необходимые данные подставляются в расчетные формулы, получаются результаты расчета каждого ВР в валюте учета, полученные результаты записываются в РР.

Основная таблица РР в БД хранит записи для каждого ВР по сотрудникам. Каждая запись содержит как исходные данные ВР, так и полученный результат расчета в валюте учета.

РР как минимум имеет измерение Сотрудник. Если расчет начисления требует детализировать, например, ПО ДОЛЖНОСТИ, КОГДА СОТРУДНИК СОВМЕЩАЕТ ВНУТРИ ОРГАНИЗАЦИИ

Результаты расчета хранятся в ресурсах. Когда требуется производить расчет в нескольких валютах, их может быть несколько.

Реквизиты РР - вспомогательные поля, в которые можно записывать различную вспомогательную / справочную информацию, относящуюся к конкретной записи. Некоторые реквизиты РР участвуют в работе его функционала.

Настройки регистра расчета задаются на вкладке Основные. Одному РР соответствует один ПВР. Свойство Период действия РР работает следующим образом. Когда флаг активен, РР снабжен функционалом получения количества рабочего времени планового и фактического, иначе эти данные получить будет невозможно. Если ВР ПВР, который обслуживает данный РР, используют ПД, тогда флаг должен быть установлен. Поле установки данного флага в РР становятся доступными поля График, ЗначениеГрафика и ДатаГрафика. Учет рабочего времени реализуется на следующем механизме. В непериодическом регистре сведений хранится информация о рабочем времени.

В базовом случае РР содержит измерение Дата и Ресурс Значение. Таким образом на каждую дату расчетного периода формируется запись, значение измерения которой - календарная дата, а значение ресурса - длительность рабочего времени этой даты.

Рабочее время может исчисляться в различных единицах измерения (днях, часах и пр.) Когда РР связан с регистром - календарем, он получит количество рабочих дней за заданный период, отобрав записи с датами, входящими в указанный период, и, сложив цифры в ресурсе Значение.

В РР в свойство График необходимо задать имя такого регистра сведений, свойству Значение РР следует присвоить значение суммируемого ресурса (для подсчёта времени). В свойство ДатаГрафика -установить измерение регистра сведений, в котором находятся календарные даты.

Чаще всего сотрудники организации работают по разным графикам, поэтому в регистре сведений (календаре) создано измерение ГрафикРаботы (для хранения ПД и ФПД для каждого графика работы отдельно)

Свойство РР БазовыйПериод по своей сути аналогично свойству ПериодДействия, но касается функционала получения суммы базы. Когда флаг активен, РР обладает функционалом получения суммы базы, в противном случае - такого функционала не будет.

Если в РР рассчитываются ВР, зависящие от баз, данный флаг необходимо устанавливать в значение Истина.

Свойство Периодичность РР позволяет выбрать из списка значений периодичность РР. Месячные начисления рассчитываются в РР с периодичностью Месяц, квартальные начисления в РР с периодичностью Квартал. Таким образом, ВР следует компоновать и по периодичности начислений.

## 7) Настройка Измерений, Ресурсов и Реквизитов РР:

Состав Измерений - набор ключевых полей, относительно значений которых информация хранится в регистре. В РР обязательным измерением является Сотрудник. Если сотрудник работает одновременно на нескольких должностях или исполняет несколько обязанностей одновременно постоянно или в течение определенного периода, тогда необходимо вводить новое измерение в РР (Должность или Подразделение), чтобы зарплату сотруднику можно было рассчитать в разрезе этого измерения.

Ресурсы - целевые поля регистра, в них хранится информация, относящаяся к комбинациям значений измерений, а также функционал регистра получает на основании значений этих ресурсов результирующие показатели. Рекомендуемая точность данного значения типа Число равна 2.

Реквизиты - вспомогательные поля, которые участвуют в работе функционала регистра только в частных случаях. В реквизитах хранится информация, которая нужная при расчете, когда система работает с конкретной записью регистра. Часто в реквизитах хранится опорная информация, смысл и содержание которой различен для каждой формулы расчета. Например, для формулы расчета оклада в реквизите может храниться месячная ставка оклада, для формулы расчета премии - это процент премии, который берется от суммы расчетной базы, при этом значение реквизита чаще всего пари расчете заполняется программно, при формировании предварительного набора записей РР. Такие значения хранят в записи РР для удобства подстановки в расчетную формулу.

Реквизит ГрафикРаботы:

Вариантов графика работы может быть много, они отличаются как количеством рабочих дней так и количеством рабочих часов, и они могут быть разными для отдельных сотрудников в разное время и для различных ВР. При наличии более одного графика система предлагает следующий подход получения количества рабочего времени.

В регистр сведений добавляется измерение ГрафикРаботы, которое указывает вариант графика, к которому относится строка регистра - календаря. В этом случае для одной календарной даты в регистре - календаре будет не одна запись, а несколько - по числу используемых графиков работы. Для того, чтобы можно было передать значение графика работы механизму расчета, в регистре расчета создается реквизит со специальной настройкой. Для того, чтобы реквизит участвовал в работе механизма расчета, необходимо заполнить свойство СвязьСГрафиком этого реквизита, а именно в нем нужно указать имя измерения регистра - календаря, в котором содержатся значения графиков.

Понятие "Данные графика":

После того, как введено понятие графика работы и настроен регистр расчета для работы по нескольким графикам, необходимо ввести количество планового рабочего времени, получаемого по периоду действия , и количества фактически отработанного времени, получаемого по фактическому периоду действия, обобщающее определение - ДанныеГрафика.

## 8) Этапы сеанса расчета зарплаты:

1.Сначала для расчета необходимо получить исходные данные(данные документа в РР):

Для расчета необходимы исходные данные, которые являются операндами формул расчета начислений. Для получения необходимых данных нужно обратиться к регистру расчета через специальный интерфейс. Чтобы сформировать и выдать программе необходимые данные, РР должен иметь значения исходных данных - границы периода действия, границы базового периода действия, значение графика работы, на основании которых формируем необходимые данные. Например, количество отработанного времени рассчитывается в интервале периода действия. Функционал РР может получить исходные данные только из полей записей РР. Для этого служат стандартные поля ПериодДействияНачало, ПериодДействияКонец и пр. Также для целей передачи значений графика работы разработчик создает реквизит ГрафикРаботы и указывает в конфигураторе, что этот реквизит содержит значение графика работы.

Таким образом, прежде чем обратиться к РР для получения необходимых данных, в РР должны быть записаны подлежащие расчету строки с исходными данными. Первый этап - формирование записей РР со стандартными полями, содержащими исходные данные.

2.После того, как записи со стандартными полями сформированы, требуется получить необходимые данные. Для этого следует обратиться к функционалу регистра расчета, эти данные разработчик получает в результате выполнения запроса.

Необходимые данные получают для каждой записи регистра расчета, которая была создана в результате выполнения первого этапа, причем так, что программа сможет в процессе расчета результатов для каждой записи регистра, которая рассчитывается, найти соответствующую ей запись в полученной из запроса таблице необходимых данных (по номеру строки).

3. Следующий этап состоит в последовательном обходе записей РР и применении расчетной формулы, соответствующей каждой записи(Чтение записей из РР, расчет нужных полей, дополнение записей РР и запись уже измененных движений в РР, чаще всего связь основной таблицы с виртуальной по НомеруСтроки). Для каждой записи устанавливается соответствие со строкой таблицы необходимых данных, полученной на предыдущем этапе. Из этой строки таблицы необходимых данных извлекаются значения необходимых данных и подставляются в выражение расчетной формулы. Полученный результат (рассчитанная сумма) помещается в ресурс записи РР, после этого записи с рассчитанными результатами снова записываются в ИБ. Таким образом на данном этапе получены рассчитанные суммы.

Поскольку в большинстве случаев при расчете начислений имеются зависимости, то этапы 2 и 3 повторяются не один раз. а по числу уровней зависимостей.

## 9) Особенности метода Записать набора записей РР

Метод «Записать» набора записей РР содержит два входных параметра:

1.Флаг «Замещать» для замещения прежнего набора записей

2.Флаг «ТолькоЗапись» для указания будет ли набор записей записан без дополнительных действий (Истина) или при записи будут определены состав и границы интервалов фактического периода действия записей набора (Ложь).

При выполнении первого этапа процесса расчета зарплаты значение второго параметра - Ложь, поскольку после записи исходных данных будет выполняться второй этап сеанса расчета, на котором программа обращается к РР для получения данных о фактически отработанном времени. Актуальность данных обеспечивается тем, что предварительно получают полный состав интервалов фактического периода действия, соответствующий составу вытесняющих видов расчета, введенных в документе.

Поэтому при выполнении первого этапа расчета необходимо установить этот параметр в значение Ложь. При выполнении третьего этапа расчета не будет необходимости рассчитывать состав и границы интервалов фактического периода действия - период действия в расчете не меняется, значит и фактический период действия (ФПД) не изменится. Поэтому при записи набора с готовыми результатами расчета начислений значение параметра ТолькоЗапись будет установлено в значение Истина, поскольку на этом этапе будет произведена запись набора без дополнительных действий.

Для выполнения первого этапа сеанса расчета параметр ТолькоЗапись устанавливаем в значение Ложь, для выполнения третьего - в Истина. Процедура пересчета ФПД потребляет много ресурсов и является длительной. Это является одной из причин необходимости установки параметра ТолькоЗапись метода Записать в значение Иситина - значение этого параметра по умолчанию - Ложь, потому на первом этапе расчета его можно не указывать, а на третьем - указать нужно.

## 10) Виртуальная таблица РР ДанныеГрафика:

Данные графика - обобщающее название для двух операндов расчетной формулы: ЧислоОтработанныхДней, ЧислоРабочихДнейВМесяце, поскольку они должны быть получены с учетом заданного графика работы.

Виртуальная таблица ДанныеГрафика содержит следующие поля:

Она воспроизводит полностью всю структуру регистра расчета - своего "хозяина": от ПериодРегистрации до ГрафикРаботы.Помимо них в таблице содержатся поля результирующих показателей виртуальной таблицы : ЗначениеПериодДействия, ЗначениеФактическийПериодДействия, ЗначениеБазовыйПериод, ЗначениеПериодРегистрации.

Эти поля представляют собой количество рабочего времени по ресурсу Значение регистра - календаря ГрафикиРаботы, с которым связан РР по четырем периодам, названия которых видны во второй части названия каждого поля.

Для наших целей поля количества рабочего времени по периоду регистрации и по базовому периоду не нужны, а нужны только поля ЗначениеФактическийПериодДействия и ЗначениеПериодДействия, которые дадут для каждой записи регистра значения операндов расчетной формулы ЧислоОтработанныхДней и ЧислоРабочихДнейВМесяце.

Данная виртуальная таблица содержит параметр Условие, в котором следует задавать условия предварительного отбора. Расчет записей происходит только для выбранного документа, поэтому нужен отбор по регистратору, а также в случае наличия категории расчетов и по категории расчета, поскольку за один этап расчетного алгоритма можно получить результаты видов расчета только одного вида расчета.

Кроме этого следует выбрать поле НомерСтроки данной таблицы, поскольку это поле является связующим полем записи результата виртуальной таблицы с записью РР, которой соответствует эта запись виртуальной таблицы. Поле НомерСтроки - ключ соответствия записи регистра и ее необходимых данных.

Для получения результата для каждой строки набора записей РР нужно:

1.Установить расчетную формулу для этой строки (для этого у каждого ВР имеется реквизит СпособРасчета).Это полезно, когда формулы расчета у ВР отличаются.

2. Найти соответствующую ей строку в таблице результата запроса - т.е. в таблице необходимых данных, которые сформировал механизм РР.

3. Прочитать из найденной строки содержащиеся в ней необходимые данные и подставить в расчетную формулу для получения результата расчета для рассчитываемой строки регистра.

Установка соответствия для строки набора записей достигается за счет поля НомерСтроки. Это поле соответствия строки набора записей и строки необходимых данных, полученных для нее. Найти строку необходимых данных, соответствующую записи РР, значит найти в ней такую строку, значение поля НомерСтроки которой равно значению поля НомерСтроки записи РР.

## 11) Подход к установлению соответствия строк регистра и строки необходимых данных

Подход к установлению соответствия строк регистра и строки необходимых данных

Поскольку выгрузка данных в таблицу значений является ресурсоёмким, целесообразно для установления соответствия строк использовать один из приемов. Для него следует упорядочить результат запроса по НомеруСтроки:

А. Поиск по индексу набора записей

1. Обходить в цикле выборку результата запроса необходимых данных

2. Вычислять на основании значения поля НомерСтроки каждой позиции выборки индекс соответствующей записи в коллекции набора записей ( индекс записи в коллекции равен [НомерСтроки -1]) и получать запись по индексу из коллекции.

Обход в цикле не набора записей регистра, а выборки запроса необходимых данных обоснован тем, что запрос содержит персистентную (основную, исходную ) таблицу. стоящую слева, и номера строк по этой причине будут те же самые, что и в наборе записей от 1 до N.

Пока Выборка.Следующий() Цикл

// получаем запись регистра из набора записей

ЗаписьРегистра = НаборЗаписей[Выборка.НомерСтроки - 1];

//реализация расчетных формул для каждого способа

КонецЦикла

В результате программа имеет доступ к полям рассчитываемой записи и к позиции выборки , содержащей необходимые данные для расчета этой записи - есть доступ ко всем операндам формулы.

Б. Поиск с использованием метода НайтиСледующий объекта ВыборкаИзРезультатаЗапроса

По своей сути прием похож на Поиск по индексу набора записей, но соответствие устанавливается посредством структуры.

Выборка = Результат.Выбрать();

//обходим все строки коллекции движений РР

Для Каждого Движение Из Движения Цикл

//сбрасываем выборку

Выборка.Сбросить();

//задаем структуру поиска

СтруктураПоиска = Новый Структура("НомерСтроки", Движение.НомерСтроки);

//ищем в выборке строку, соответствующую строке Набора движений

Если Выборка.НайтиСледующий(СтруктураПоиска) Тогда

//реализация расчетных формул для каждого способа

КонецЕсли;

КонецЦикла;

## 12) Виртуальная таблица РР ФактическийПериодДействия

Виртуальная таблица ДанныеГрафика дает количество отработанного времени, суммируя рабочие дни по всем интервалам ФПД. Может возникнуть ситуация, когда расчетная формула потребует доступ не только к количеству рабочего времени за ФПД, но и множество самих интервалов ФПД, т.е. их количество и границы этих интервалов.Границы интервалов ФПД можно получить запросом, обращаясь к виртуальной таблице ФактическийПериодДействия РР. Эта таблица даёт список интервалов ФПД. Структура ее полей полностью повторяет структуру полей РР. Каждая запись этой таблицы соответствует одному интервалу ФПД РР , а границы этого интервала находятся в полях ПериодДействияНачало и ПериодДействияКонец записи. Для каждого вида расчета эта таблица содержит столько записей, на сколько интервалов был разбит изначальный период действия под воздействием вытеснения.

Эта таблица принимает только один (необязательный) параметр - Условие, представляющий собой логическое выражение условия предварительного отбора. Имеет смысл задать условие отбора по сотруднику.

## 13) Виртуальная таблица База РР

При решении расчетных задач возникает необходимость получения исходных для расчета данных типа база. Иначе эти данные можно описать как сумму базовых начислений для тех видов расчета, в расчетных формулах которой присутствует этот операнд.

Например, необходимо вычислить надбавку по следующей формуле:

Надбавка = СуммаБазовыхНачислений \* ПроцентПремии / 100

ПроцентПремии - значение целочисленного типа , которое целесообразно хранить в реквизите Размер. СуммаБазовыхНачислений - значение, которое следует получить, используя функционал РР. Для данного ВР помимо диапазона действия базового периода приходится задавать диапазон действия периода действия поскольку этот ВР находится в ПВР, у которого активен флаг ПериодДействия. Поскольку в РР имеется такая настройка, поля ПериодДействияНачало и ПериодДействияКонец необходимо задавать в каждой его записи, иначе набор записей в информационную базу записан не будет.

Если в метаданных РР установлено свойство Период действия, то в каждой записи регистра обязательно требуется задавать границы периода действия, даже если расчетная формула для этой записи не требует получения данных графика. Если это требование системы не выполняется, набор записей не будет записан в ИБ.

Получение суммы базы реализовано путем обращения к виртуальной таблице База РР.

Следует отметить, виртуальных таблиц База у РР может быть несколько. Количество этих таблиц равно количеству базовых РР, т.е. таких РР, в которых у данного РР могут находиться результаты расчета базовых ВР.

Число базовых РР как правило не менее числа базовых ПВР у того ПВР, с которым связан данный РР. Это обусловлено тем, что каждый ПВР может быть связан с несколькими РР.

У РР таблиц База может быть несколько - по числу его базовых РР.

Виртуальная таблица База именуется следующим образом: База + имя одного из базовых регистров РР - "хозяина".

Рассмотрим механизм получения базы на примере. ВР Надбавка хранится в ПВР ОсновныеНачисления вместе с ВР Оклад и рассчитывается как процент от начисленного в данном периоде Оклада.

Запрос строится на основе персистентной таблицы РР ОсновныеНачисления и выберем из нее поля НомерСтроки и ВидРасчета.СпособРасчета . Наложим условия по Регистратору и КатегорииРасчета, зададим упорядочивание по НомеруСтроки. Добавим в список таблиц запроса виртуальную таблицу ОсновныеНачисления.БазаОсновныеНачисления. У этого РР одна таблица базы, т.к. у него один базовый регистр, в данном случае - он сам .

На закладке Связи необходимо создать левое соединение таблиц по НомеруСтроки.

## 14) Параметры виртуальной таблицы База

Форма параметров виртуальной таблицы содержит 4 поля :

1.ИзмеренияОсновногоРегистра

2.ИзмеренияБазовогоРегистра

3.Разрезы

4.Условие

В параметре Условие следует сделать отбор по Регистратору и по КатегорииРасчета.

Параметры ИзмеренияОсновногоРегистра и ИзмеренияБазовогоРегистра нужно задавать обязательно. Это - единственный случай виртуальных таблиц, когда параметры виртуальной таблицы являются обязательными.

Виртуальной таблице для получения суммы базы надо брать только те записи базового регистра, в которых значения измерений совпадают со значениями соответствующих измерений регистра, для которого получается сумма базы. При этом имена измерений в базовом регистре могут отличаться от имен измерений регистра, для которого получается сумма базы.

Для того, чтобы установить соответствие имен измерений регистров служат параметры ИзмеренияОсновногоРегистра и ИзмеренияБазовогоРегистра. Эти параметры следует задавать объектами типа Массив или СписокЗначений, причем в Консоли запросов доступен только СписокЗначений.

В параметре ИзмеренияОсновногоРегистра в качестве элементов размещаются строковые имена измерений основного регистра, в параметре ИзмеренияБазовогоРегистра - базового регистра, причем в том же порядке. Таким образом механизм виртуальной таблицы сопоставляя элементы с именами основного и базового регистра может установить, какое измерение базового регистра какому измерению основного регистра соответствует.

Параметры ИзмеренияОсновногоРегистра и ИзмеренияБазовогоРегистра должны задаваться обязательно. даже. когда измерения основного и базового регистра совпадают.Значений по умолчанию для них нет.. В случае совпадения измерений можно установить в эти параметры одно и то же значение.

## 15) Состав полей виртуальной таблицы База :

Виртуальная таблица РР База содержит поля РР - "хозяина", набор полей с суффиксом "Разрез" и результирующее поле таблицы "РезультатБаза". Результирующие поля рассчитываются по всем ресурсам базового регистра, а названия состоят из имени ресурса и суффикса "База".

## 16) Использование периода действия в качестве базового периода

Бывают случаи, когда ПериодДействия в расчете начисления используется вынужденно (из -за настроек ПВР, а именно активного флага ПериодДействия - он нужен для некоторых ВР), Возможно, в некоторых случаях стоит разделять ВР по ПВР в соответствии с этим признаком.

Если же такое разделение невозможно по ряду причин, тогда можно избежать ввода одинаковых значений границ интервала периода действия и базового периода. С одной стороны ввод границ базового периода не контролируется платформой, с другой стороны - у ВР, принадлежащих ПВР есть стандартный реквизит ПериодДействияБазовый, который имеет тип Булево и на форме ВР обозначен подписью БазовыйПериодКакПериодДействия. Если этот реквизит установить в значение Истина, тогда механизм расчета получения суммы базовых начислений будет использовать заданные границы периода действия , как границы базового периода.При этом значения. заданные в полях БазовыйПериодначало и БазовыйПериодКонец будут игнорироваться.

Если для ВР, который требует получения суммы базовых начислений, установить свойство ПериодДействияБазовый в значение Истина, то для границ базового периода можно будет использовать период действия , а поля границ базового периода - оставлять пустыми.

## 17) Расчет суммы начисления , зависящей от нескольких базовых начислений

Рассмотрим пример, когда необходимо реализовать следующий состав видов расчета:

1. Премия персональная = Фиксированная сумма;

2. Месячная премия = (ПерсональнаяПремия + Оклад + ОкладНаВыезде) \*ПроцентПремии /100 - начисляется за месяц

3.Поощрительная надбавка = МесячнаяПремия \* ПроцентНадбавки /100 - начисляется за месяц

4. Премия за период = Оклад \* ПроцентПремииЗаПериод / 100 - начисляется за переменный период в пределах месяца

В данных ВР ни один из ВР не зависит от планового или фактического периода действия, но имеются множественные зависимости от базовых начислений. Создадим новый ПВР для реализации поставленной задачи. Поскольку ни одна из расчетных формул не использует в качестве операнда данные графика, то этот ПВР не будет использовать период действия, значит вытесняющие ВР для него отсутствуют.

Поскольку все ВР являются начислениями, то зависимость от базы будет по периоду действия.Однако в данном ВР период действия не используется, а это значит, что период действия равен периоду регистрации. То есть если у базового ВР отсутствует период действия, то при определении попадания его результата в сумму базовых начислений считается, что период действия равен периоду регистрации.

У ВР МесячнаяПремия базовые ВР находятся в разных ПВР: ПерсональнаяПремия в премиях, а Оклад и ОкладНаВыезде в ОсновныхНачислениях. Значит, нужно оба ПВР указать в качестве базовых для ПВР Премии.

Новый РР предназначен для расчета начислений, которые содержатся в ПВР Премии.Ни один из ВР не требует для реализации расчетной формулы данных графика, следовательно флаг ПериодДействия неактивен. ВР из ПВР Премии рассчитываются , в основном, с использованием расчетной базы, следовательно флаг БазовыйПериод активен. Регистр рассчитывает месячные премии, значит, значение поля Периодичность будет Месяц. Прикладные поля регистра расчета это - измерения (Сотрудник, Должность), Ресурс (Результат) и Реквизит (Размер)/

ВР ПерсональнаяПремия простой, и не требует пояснений. Для всех остальных ВР из ПВР Премии два регистра базы: результаты базовых ВР выбираются из РР ОсновныеНачисления и из РР Премии. В запросе для получения необходимых данных каждой записи документа в РР Премии должны соответствовать строки двух виртуальных таблиц База : база по регистру ОсновныеНачисления и база по регистру Премии(Премии.БазаОсновныеНачисления, Премия.БазаПремии).Нужно в запросе получить таким образом соотнесенные записи исходной (персистентной) таблицы регистра Премии для документа и соответствующие им записи обеих таблиц. Один из способов построения такого запроса состоит в использовании операции СОЕДИНЕНИЕ языка запросов: условием соединения будет равенство полей НомерСтроки исходных таблиц запроса. Важно учесть, что виртуальные таблицы регистров платформы не возвращают записи с нулевыми результатами. Например, если для записи РР по базовому регистру сумма базы равна 0, то виртуальная таблица не вернет строки с этим номером. Значит, что запрос для получения суммы базы по нескольким базовым регистрам нужно строить с учетом этого факта, иначе при неправильном применении соединения можно потерять сумму базы по какому - либо базовому регистру. Чтобы это учесть применяют следующий прием : в запросе выбирают персистентную таблицу регистра , поскольку она содержит данные всех записей документа. С этой таблицей левым соединением соединяют виртуальные таблицы "База" тех РР, которые для рассчитываемого РР содержат базовые суммы.

## 18) Частичное попадание периода действия базового вида расчета в базовый период

В ПВР (18)ПР:) был введен ВР ПремияЗаПериод. Способ расчета у него Процентом, базовый ВР - Оклад, он является зависимым ВР первого уровня.

Этот ВР частично укладывается в период действия базового ВР. Если установить базовый период действия не месяц, а несколько дней, сумма базовых начислений уменьшится пропорционально уменьшению количества рабочего времени в базовом периоде, иначе будет получен несправедливо большой результат расчета по базе. Так происходит с суммой базовых начислений по тем базовым ВР, которые обладают периодом действия, т.е. по формуле расчета зависят от количества отработанного времени. Когда базовый период частично попадает в границы периода действия базового ВР, то сумма базовых начислений считается пропорционально кол-ву рабочего времени в базовом периоде. Это утверждение справедливо только для тех базовых ВР, которые обладают периодом действия, т.е. зависят по расчетной формуле от отработанного времени.

Это не распространяется на базовые ВР, которые не обладают периодом действия.

## 19) Получение суммы базы в разрезе базовых ВР:

Параметр Разрезы виртуальной таблицы База позволяет развернуть сумму базовых начислений любого ВР в разрезе указанных полей, а также в разрезе комбинаций этих полей. Например, если интересует разрез суммы базы по базовым ВР, т.е. сколько каждый базовый ВР внес в общую сумму базового начисления. Это позволяют сделать указанные средства настройки виртуальной таблицы База РР. Для этого нужно в запросе выбрать одно или несколько полей этой таблицы, имеющих суффикс Разрез в названии. Нужна та комбинация полей, по которой нужно развернуть сумму базы для каждого ВР. Однако, этого недостаточно, нужно задать в запросе виртуальной таблицы "База" параметр - "Разрезы". Этот параметр задается объектом типа Массив или СписокЗначений. Элементами массива являются имена полей - разрезов без суффикса Разрез, по которым необходимо развернуть сумму базы.

Возможность получать сумму базы в разрезе дает довольно ценную информацию для учета, а также для выяснения отношений с сотрудниками по суммам начислений и удержаний. например, сотрудник пришел в бухгалтерию по вопросу неверного начисления премии. Получив сумму базовых начислений для премии в разрезе базовых видов расчета премии, можно детально разобраться в этом вопросе.

## 20) Построения отчета по начислениям.

Когда отчет по начислениям необходимо построить в разрезе сотрудников, их должностей, регистров расчета и видов расчета, следует в СКД использовать набор данных типа Объединение. При этом, чтобы иметь возможность образовывать группировку отчета по имени регистра расчета в запрос вводят параметр, в котором будет задаваться строковое название РР.

## 21) Корректировка результатов расчета за прошедший период

Бухгалтерская отчетность составляется со строгим соблюдением установленной периодичности, при этом отчетность, предоставляемая в контролируемые органы датируется периодом регистрации. Сданная бух. отчетность корректировке и изменению не подлежит. Из этого следует, что, если в отчетности за прошедший период была допущена ошибка, и это заметили в текущем периоде, то повторная подача отчетности за прошлый период невозможна, поскольку в закрытый период изменения внести нельзя. Для решения подобных задач используется специальный подход.

Важно понимать, если период регистрации вытесняющего вида расчета больше (позже) его периода действия, то расчетные механизмы запрещают вытеснение, то есть запрещают изменение результатов расчета закрытого периода.

Механизм сторно (формирование записи (проводки с той же корреспонденцией счетов), но с отрицательной суммой) позволяет решить задачу корректировки в текущем периоде записей прошлых периодов. В общем случае периоды, за которые нужно рассчитать сторно - суммы заранее не известны. Платформа дает средство получения всей необходимой информации , в т.ч. и периодов сторнирования, для расчета сторнируемых сумм.

Метод объекта РР ПолучитьДополнение() возвращает объект ТаблицаЗначений, каждая строка которой соответствует ситуации, когда платформа запрещает вытеснение по причине того, что период действия вытесняющего ВР меньше периода регистрации.

В таблице значений столько строк, сколько раз платформа запрещает вытеснение, а каждая строка содержит всю необходимую информацию для расчета сторно - суммы.

Значит, перед расчетом сторно - суммы следует вызывать метод ПолучитьДополнение и обработать полученную таблицу значений, создавая для каждой ее строки сторно - запись РР. Разместить вызов этого метода следует , когда система определит состав интервалов фактического периода действия для каждого вида расчета. Это происходит , когда система выполняет запись набора движений с исходными данными - когда записываются данные в базу с переопределением ФПД. Если к этому момент получен актуальный ФПД - результат вытеснения, то, РР к этому моменту имеет информацию о случаях, когда вытеснение было запрещено. Следовательно, в данном месте и нужно вызывать метод ПолучитьДополнение() с последующей обработкой результата.

Таблица сторно содержит повторение строки того ВР, для которого была сделана попытка вытеснения и поля

1.ПериодРегистрацииСторно(период регистрации того ВР, который попытался вытеснить ),

2. ПериодДействияНачалаСторно, ПериодДействияКонецСторно - начало и конец периода действия этого ВР

Таким образом для расчета сумму сторнируемого начисления необходимо:

1. Создать запись РР с исходными данными, соответствующими полученным в строке таблицы, но период регистрации будет таким, который указан в поле ПериодРегистрацииСторно, а границы периода действия будут такими, какие указаны в полях строки таблицы ПериодДействияНачалоСторно и ПериодДействияКонецСторно.

2. Рассчитать результат этой записи.

3. Изменить знак полученного результата записи на отрицательный.

По данным каждой строки нужно создать новую запись в РР, заполняя ее поля значениями из полей строки таблицы сторно - записей, затем снова записать набор записей и рассчитать его (записать с пересчетом ФПД, поскольку новые записи РР могут оказать влияние на ФПД). Флаг Сторно в каждой строке устанавливается в значение Истина, это - признак для процедуры расчета общего модуля, что знак результата надо изменить на отрицательный., то есть:

Запись. Результат = Результат \* ?(Запись.Сторно, -1,1);

Кроме того, может потребоваться добавлять сторно - запись в табличную часть документа расчета начислений.

## 22)Перерасчеты

Зависимость по перерасчету:

Результаты ВР часто зависят от начислений по другим ВР. Для того, чтобы в случае перерасчета начисления по ВР, от которого зависят начисления по другим ВР, можно было автоматически определить те начисления , которые нужно пересчитать в связи с изменениями используется механизм Перерасчетов.

Если после перерасчета ВР А необходимо перерассчитать ВР Б, то ВР Б зависит по перерасчету от ВР А.

Зависимость по перерасчету может быть как прямой так и опосредованной (неявной/ косвенной). Зависимость по перерасчету шире зависимостей по ПД или по Базе.

На примере разберем порядок построения механизма получения запросом необходимых данных для выполнения перерасчетов. Настройка зависимости по перерасчетам осуществляется в ПВР, в табличной части ВедущиеВидыРасчета. Эта табличная часть присутствует в ПВР всегда, ее наличие не зависит от настроек метаданных ПВР. ПВР всегда имеет от 1 до 3 стандартных ТЧ. ТЧ ВедущиеВидыРасчета можно заполнять как в Конфигураторе таки в режиме 1С:Предприятие. Базовые ВР помещаем в строки ТЧ ВедущиеВидыРасчета.

## 23) Объект конфигурации Перерасчет

Объект Перерасчет логически подчиняется объекту РегистрРасчета, находится среди метаданных РР в дереве конфигурации. Объект Перерасчет - таблица ИБ, т.е. персистентный объект. Работа платформы с перерасчетом происходит так. Когда происходит перерасчет начислений ВР А, который находится в табличной части ВедущиеВидыРасчета ВР Б, тогда платформа помещает строку с ВР Б и сопутствующей информацией в таблицу перерасчета. Подразумевается, что прикладная программа должна прочитать данные таблицы перерасчета и перерассчитать ВР Б.

Платформа по умолчанию ничего не перерассчитывает, а дает исходную информацию о ВР, которые надо перерассчитать, работает по принципу уведомлений, как и метод получить дополнение в случае получения сторно -записей.

Структура таблицы ИБ объекта Перерасчет

Объект Перерасчет содержит следующие поля:

ВидРасчета, ОбъектПерерасчета - ( эти поля создаются автоматически), набор измерений.

ВидРасчета - это ВР, который надо перерасчитать, а ОбъектПерерасчета - ссылка на документ - регистратор, который сделал в РР движение с данным ВР.

ОбъектПерерасчета - ссылка на документ - регистратор, который сделал в РР движения с данным ВР

Пересчитывать ВР нецелесообразно, поэтому необходимо создавать набор уточняющих измерений, который соответствует набору измерений РР - хозяина перерасчета. Например, перерассчитываем не просто ВР Надбавка, а надбавка для сотрудника, который работает в должности (то есть нужно создать измерения Сотрудник и Должность). Количество измерений в РП будет равно кол-ву измерений РР - хозяина.

После того, как измерения ПР заданы, необходимо установить соответствие измерений перерасчета измерениям РР.

Имена измерений перерасчета не обязательно совпадают с именами измерений РР - хозяина. Поэтому необходимо установить соответствие измерения перерасчета измерению РР- хозяина, а также измерениям ведущих РР, т.е. таких, в которых хранятся результаты ведущих ВР для данного расчета. Это делается через настройку свойств измерения перерасчета. В свойстве ИзмерениеРегистра надо выбрать из списка измерение РР - хозяина перерасчета, которому соответствует созданное измерение перерасчета. А в свойстве ДанныеВедущихРегистров надо указать флагами соответствие этого измерения перерасчета с измерениями всех ведущих РР для данного РР. Форма структуры ведущих РР открывается нажатием на кнопку выбора в поле свойства ДанныеВедущихРегистров. Например, у РР Премии результаты ведущих ВР находятся в РР ОсновныеНачисления и Премии, поэтому флажки измерений нужно установить у измерения обоих РР.

## 24) Объектная модель работы с перерасчетом

Объектная модель Перарасчета по структуре схожа с объектной моделью любого регистра. Имеется объект МенеджерПерерасчета, объект ПереасчетНаборЗаписей, который подобно набору записей регистра является коллекцией объектов ПерерасчетЗапись. Объект ПерасчетНаборЗаписей имеет метод Добавить, который позволяет программно создать новую запись перерасчета и заполнить ее свойства, метод Записать для записи наборов перерасчета в ИБ, метод Удалить для удаления записи из перерасчета.

Этот объект также позволяет программировать обработчики событий , а состав событий аналогичен составу событий для регистра. Модуль набора записей перерасчета, где можно описать обработчики событий открывается из палитры свойств перерасчета в Конфигураторе. Таким образом не только платформа, но и разработчик может изменять состав и содержание записей перерасчета. Это необходимо для случаев когда, например сумма начисления рассчитывается в зависимости от некоего оборотного показателя, например объема продаж за месяц. Эти данные хранятся в оборотном РН, следовательно механизм перерасчета не может быть в курсе изменения суммы сделки, которое могло произойти при проведении документа - регистратора по РН оборотов. Значит, запись в таблицу перерасчета для пересчета результата ВР автоматически не попадет , ее нужно сформировать программно, в момент после перепроведения документа по РН.

## 25) Две особенности реализации Перерасчетов

1.При программной реализации перерасчетов необходимо пересчитывать не весь документ, который указан в поле ОбъектПерерасчета ПР, а только те виды расчета, которые попали в Перерасчет.

2.Таблица Перерасчета идентифицирует не запись РР, которую надо перерассчитать, а только ВР. Если таких ВР в наборе несколько, то пересчитать их надо все. Следует помнить, что для сотрудника в РР может быть несколько записей с одинаковым ВР и на основании единственности записи в таблице перерасчета, нужно пересчитать все записи с данным ВР, а не одну.

В большинстве случаев запрос для получения необходимых данных строится с помощью левого соединения персистентной таблицы, которая обеспечивает состав строк набора записей рассчитываемого документа и способ расчета, применяемый к каждой строке, с виртуальными таблицами по НомеруСтроки. Состав номеров записей регистра, подлежащих расчету, и способов расчета для этих записей задает именно эта персистентная таблица. Следовательно, если эта таблица содержит не все записи набора, рассчитываемого документа, а только подлежащие перерасчету, то такой запрос позволяет реализовать условие первой особенности 1.

Такую таблицу можно получить с помощью выборки из таблиц перерасчетов, которую следует соединить с персистентной таблицей РР по равенству полей ВидРасчета в обеих таблицах и по равенству полей Регистратор и ОбъектПерерасчета соответственно в таблице РР м ПР, а также всех остальных измерений. Результат такого запроса даст состав тех записей РР, которые подлежат перерасчету с их способами расчета. Эту таблицу можно использовать в качеств задающей таблицы для получения необходимых данных, затем следует соединить данную таблицу с виртуальной таблицей базовых начислений и завершить процедуру перерасчета записей.Полученный результат можно передать в процедуру РассчитатьЗаписиРегистраРасчета, которая была разработана ранее и использовалась для расчетов

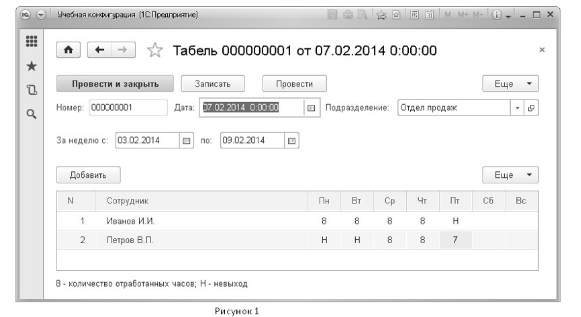
Поскольку запрос возвращает состав записей регистр, подлежащих пересчету, а содержание результата запроса по структуре полей аналогично содержанию результата запроса для выполнения основного расчетного алгоритма, то становится ясно, что при таком подходе построения расчетного алгоритма, алгоритмы расчета и перерасчета отличаются только текстом запроса. Это эффективно.

Следует принять во внимание, что перерасчету могут подлежать записи РР закрытого периода. Поэтому при реализации перерасчетов изменения в результаты существующих записей РР не вносятся. Вместо этого подлежащие перерасчету суммы сторнируются и создаются новые записи РР, в которых учитываются новые результаты. По этой причине целесообразно разработать в конфигураторе специальный документ, который будет выполнять перерасчеты. Документ будут получать из таблиц перерасчетов записи, подлежащие перерасчету, с суммами результатов, формировать в наборе записей сторнирующие записи и создавать новые записи, в которые будут помещать результаты перерасчетов.

## 26) Расчет зарплаты по табелю

Задача начисления зарплаты по табелю

Начисление зарплаты сотрудникам предприятия осуществляется ежемесячно. Каждый сотрудник может работать одновременно в нескольких подразделениях компании, то есть совместительство допускается. Все сотрудники работают по графику работы, установленному для каждого подразделения отдельно. Количество фактически отработанных часов вводится в систему с помощью документа «Табель». Документ должен заполняться еженедельно на список сотрудников только определенного подразделения. Для каждого сотрудника, на каждый день недели, вводится количество фактически отработанных часов или информация о невыходе без уважительной причины. Внешний вид формы табеля представлен на следующем рисунке 1



Сотрудники предприятия получают оплату по окладу пропорционально отработанному времени в часах. Часовая ставка рассчитывается как начальное значение оклада, деленное на количество рабочих часов в том же периоде, что и фактически отработанные часы. Первоначальное значение оклада может изменяться не чаще, чем один раз в день, но берется на начало расчетного периода. В информационной базе необходимо хранить историю его изменения.

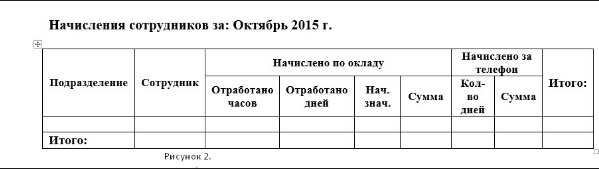
За каждый отработанный день в течение периода начисления сотрудникам предприятия полагается фиксированная сумма денег в качестве компенсации затрат на разговоры по мобильному телефону. Размер суммы един для всех сотрудников компании и должен быть зафиксирован в информационной базе.

Невыход сотрудника на работу без уважительной причины должен быть зафиксирован в информационной базе, но не оплачен.

Механизм перерасчетов в рамках данной задачи использовать не надо.

Ввод всех начислений происходит документом «Начисление зарплаты». Документ в расчетном периоде может быть один (сразу для всех видов расчета), а может быть несколько (по одному для каждого отдельного вида расчета). Считать, что все данные вводятся только в пределах одного месяца, например, можно указать начисление оклада с 10.01 по 31.01, а запись оклад: с 10.01 по 03.02 вводить нельзя. В одном документе могут быть данные только за текущий расчетный период.В форме документа «Начисление зарплаты» необходимо предусмотреть наличие кнопки «Рассчитать», при нажатии на которую, будет произведен предварительный расчет заработной платы. Результат расчета должен быть отражен в табличной части этой же формы.

Для анализа сделанных сотрудникам предприятия начислений в конфигурации необходимо предусмотреть отчет следующего вида (рисунок 2):



Отчет должен быть построен только за определенный календарный месяц. Изменять размер периода отчета пользователь

### 26.1) Виды расчета:

Расчет зарплаты по табелю.Пройдемся по ВР:

1.Оклад = НачальноеЗначениеОклада / N раб.часов \* N ФПД, где НачальноеЗначениеОклада получаем из РС СведенияОСотрудникахСрезПоследних,

N раб.часов - из графика, N ФПД из табеля. Если в задаче есть табель, ФПД берем из табеля, а ПД из графика.

2.КомпенсацияЗаТелефон = ДневнаяСтавка \* N ФПД, где

ДневнаяСтавка - константа, N ФПД - из табеля

3.Невыход. Он должен быть зафиксирован в ИБ, значит, его нужно отражать в РР

Расставим приоритеты расчета. Невыход не имеет приоритета, ОплатаЗаТелефон тоже не требует приоритета. Настроим взаимосвязи между видами расчета:

Все указанные ВР не имеют ни базовых ВР ни Ведущих

Оклад - использует ПД, остальные ВР - нет.

Проверим условие вытеснения, определим какой ВР вытесняющий, а какой нет.

1.Допустим Невыход вытесняет Оклад. Невыход может вытеснять Оклад , если он непосредственно влияет на ФПД. Для ФПД используется табель, а не время графика, значит Невыход нет смысла делать вытесняющим.

Рассмотрим ВР КомпенсацияЗаТелефон . Из табеля вытеснений нет. Рассмотрим учет по ПД. Вытеснений нет, данные графика не учитываем .Учет по ПД отсутствует. Аналогично по ВР Невыход

Настроим ПВР поприоритетно.

Табель - вспомогательный документ, фиксирующий ФПД. В поставленной задаче Невыход должен быть в РР, т.к. это - самостоятельный алгоритм (т.к. у него есть результат, который не может быть произвольным, а равен константе). Невыход - событие, его надо отражать в РР, его надо рассчитать.

Методология с табелем позволяет перейти к методу отклонений и обратно. Настроим ВР: У оклада нет зависимости от базы, но есть использование ПД. КомпенсацияЗаТелефон не зависит от ПД, поэтому этот ВР выносят в отдельный ПВР.

Невыход тоже не использует, его можно поместить как в ПВР Начисления так и в ПВР Удержания, поскольку сумма по нему равна 0.

Чем меньше ВР, тем лучше, поэтому разместим Невыход вместе с ВР КомпенсацияЗаТелефон.

Поскольку зависимостей в данной задаче нет, перейдем к настройке Графиков, затем РР. Для Оклада нужен График. Графиков по задаче может быть много, но неизвестно сколько, поэтому добавляем измерение Подразделение.. Добавим в РС Графики измерение и доработаем обработку ЗаполнениеГрафика. Сделаем новый реквизит на форме, добавим входной параметр и доработаем код в Модуле обработки:

1.сделал отбор

2.задал график для новых записей

Можно перейти к определению РР. Стоит придерживаться догмы один ПВР соответствует 1РР.

Настроим данные РР: Ресурс в задаче точно будет один.

Измерения Сотрудник (небазовое, т.к. нет баз. начислений, связь с графиком), Подразделение (т.к. совмещение допускается, то это будет измерением, базовых начислений нет, связь с графиком нужно настроить)

Реквизиты, которые потребуются можно определить из вида отчета:

-отработано часов;

-отработано дней;

-начальное значение оклада.

Перерасчетов нет, Регистратор - Начисление з/п.

Настроим РР ДопНачисления. ПВР ДопНачисления

Измерения Сотрудник и Подразделение - связи с графиком нет

Ресурс :

* Результат

Реквизиты:

* ОтработаноДней;
* ДневнаяСтавка.

По умолчанию расчет идет в разрезе всех измерений. ВР - всегда предопределенное измерение.

Создадим документ Табель:

Структура документа:

Реквизиты :

* Подразделение;
* НачалоНедели;
* КонецНедели.

ТабличнаяЧасть: ФактическоеВремя: Реквизиты:

Сотрудник;

* Пн;
* Вт;
* Ср;
* ..;
* Сб;
* вс ;

Если в таблице нарисован месяц на 30 дней, значит будет 30 реквизитов, тип рекомендуется установить строковый, длина 2 символа, задается маска 99;Н.

Движения документа будут рассмотрены позже. Сперва создадим основную форму документа, форма документа должна иметь вид как в задании. Не забыть под таблицей поясняющую надпись. Переименуем заготовки колонок по названию дней.

### 26.2) Расчет зарплаты:

Расчет зарплаты по табелю:

Табель вводится еженедельно, а начисление ежемесячно, задавать границы недели вручную неудобно.При вводе любой даты в поле НачалоНедели и КонецНедели значение будет приводиться к значению начала и конца календарной недели. В обработчике события ПриИзменении для данных полей реализуем эту процедуру.

Возникает вопрос, будет ли документ проводиться. Для этого надо проверить, что будет с регистром. Главная идея регистров - минимизация таблиц, строк и колонок. Видно, что обрабатываются от 4 до 6 строк ( это мало). Однако недели - не кратны месяцам, а ввод реализован за неделю. Если неделя переходящая, то в табель так и вводится.При расчете будет реализован анализ переходимости недель. За период нужна сводная информация - накопление фактически отработанных часов.

Положим, что для этого нужен РС, непериодический, т.к. нужно количество отработанных дней за месяц. периодичность ввода данных определяется условием задачи (помесячно).Остается вопрос программной обработки границ. Если есть месяц и неделя (некратные периоды), будет проблема с границей. Для этого берем периодичность день. Проблемы границ нет, но скорость обработки снизится. Такое решение неоптимальное. Попробуем РН Обороты. Если в него заносить информацию по дням, проблем с границами не будет, скорость обработки подходит, т.к. обрабатывается 1 запись. Вывод: нужен РН Обороты, в котором фиксируется фактическое время (ДанныеТабеля)

Структура:

Ресурсы:

* ОтработаноЧасов (для Оклада);
* ОтработаноДней (для КомпенсацииЗаТелефон).

Измерения:

* Сотрудник;
* Подразделение.

Регистратор:

* Табель

Табель никогда не формирует записи в РР, он только фиксирует фактически отработанное время

Оперативное проведение запрещено, проведение разрешено.

Далее реализуем движения и обработку проведения. Важно учесть, что в случае невыхода , могут быть только ненулевые записи или все(с нулями). Логика обработки должна это учитывать. Желательно ввести несколько табелей и проверить совместительство. Переходим к документу НачислениеЗарплаты

Отредактируем структуру. Можно сделать одну таб.часть, тогда ВР - составной, а можно 2 таб. части, что будет удобнее.

Смотрим, хватает ли реквизитов. Реквизит Размер можно удалить, поскольку его получаем из РегистраСведений. ДополнительныеНачисления: ПД нет, значит ПДН и ПДК не нужны.

Сформируем с помощью конструкторов движения исходные данные для всех ВР, проводим расчет по приоритетам, обходим набор движений. Следует помнить, что и в РР и в РН в этой задаче должен быть отбор по сотруднику и Подразделению, а также по Регистратору.

Из РР берем номер строки и плановое время, сотрудников и отбор. Т.е. создаем набор , помещаем в временную таблицу. Создаем запрос в пакете: Берем данные графика, берем сведения об окладе из РС, не забываем про отборы (без повторяющихся). Из РН Обороты выбираем данные с отбором по сотруднику и по подразделению. Данные табеля и оклад проверяем на NULL.

Запрос отличается от стандартного подхода только получением ФПД. Получаем выборку, встаем в начало выборки, ищем по номеру строки в движениях (записях) набора нужные строки для расчета по номеру строки получаем необходимые данные, заполняем реквизиты ОтработаноЧасов, ОтработаноДней и Размер. Запишем движения без перезаписи таблиц ФПД.

Заполняем значения начальных окладов в РС и заполняем графики (предварительно введем виды графиков).

Создадим документ НачислениеЗарплаты. Если сотрудник по табелю не работает некоторые дни, то действуем следующим образом, как в методе отклонений, т.е. указываем запись с 01.07.2016 по 31.07.2016, не разбивая запись. Вытеснение сделал пользователь из табеля. Логика расчета одинаковая как при табеле, так и при методе отклонения. Т.е. вытеснение руками пользователя.

Произведем расчет начислений за телефон. Организуем цикл по записям набора ДопНачислеий. нет учета по ФПД, потому записываем без параметра.

Расчет зарплаты по табелю:

До цикла получаем дневную ставку за телефон и ФПД из табеля. Нужны обороты по табелю с отборами по Сотруднику и Подразделению. Есть два варианта:

а. Берем ТабЧасть

б. Берем основную таблицу РН

Начисления за телефон в дополнительных начислениях берем Сотрудника и Подразделение, отбор по Регистратору и ВР, т.к. в ПВР два разных ВР. Дубли надо отбросить ("Без повторяющихся" на закладке Дополнительно). В пакете берем обороты, задаем параметры, периодичность не нужна. Выбираем ОтрабДней.

Тут нет номера строки для позиционирования, но есть Сотрудник и Подразделение , по ним ищем нужные записи. Для отбора подойдет Структура для поиска, т.к. 2 поля, т.к. ввыборке их может не быть нужна проверка на их наличие.

Заполним реквизиты:

В режиме 1С Предприятие заполним константу и проверим дни.

Отразим Невыход

Переходим к задаче расчета в форме. Последовательность действий приведена в прикрепленном файле "РасчетВФорме".

Начинаем с расчета при проведении. Как расчет сделан, сам расчет перенесём в процедуру общего модуля. На практике в общем случае результат расчета получаем с помощью кода из общего модуля (один код получен из разных мест).

Создадим общий модуль ОбращениеКДанным

В общем модуле создаём экспортируемую процедуру Рассчитать(), в неё переносим расчёт. Передадим в качестве входных параметров требуемые данные: НаборДвижений, Ссылка и Отказ.

На форме создадим команду Рассчитать и добавим ее в командную панель документа НачислениеЗарплаты. В обработчик нажатия этой кнопки из реквизита Объект с помощью метода РеквизитФормыВЗначение получим объект, чтобы можно было обратиться к свойствам объекта, а именно к Движениям. Через объект получим доступ к ТЧ ОсновныеНачисления и ДопНачисления, а также к реквизиту Дата.

Формирование движений есть, запись есть, вызываем расчет.

Заполняем ТЧ сформированными наборами. Расчет в форме означает, что данные из формы пишутся в регистр. Для начала следует изменить состав ТЧ в соответствии с содержимым табл. РР ОсновынеНачисления, кроме полей автозаполнения (№ строки, Регистратор, Период ). Можно посмотреть их в обработчике проведения. Добавим (ПериодРегистрации, ПДН, ПДК, Результат, ОтработаноЧасов, ОтработаноДней, Размер)

Аналогично сделаем для доп. начислений (Результат, ОтработаноДней, Размер) Т.е. сделали расчет и отражаем все в табличной части. Такой подход нужен, чтобы на практике пользователь видел, что пойдет в РР и откорректировал.

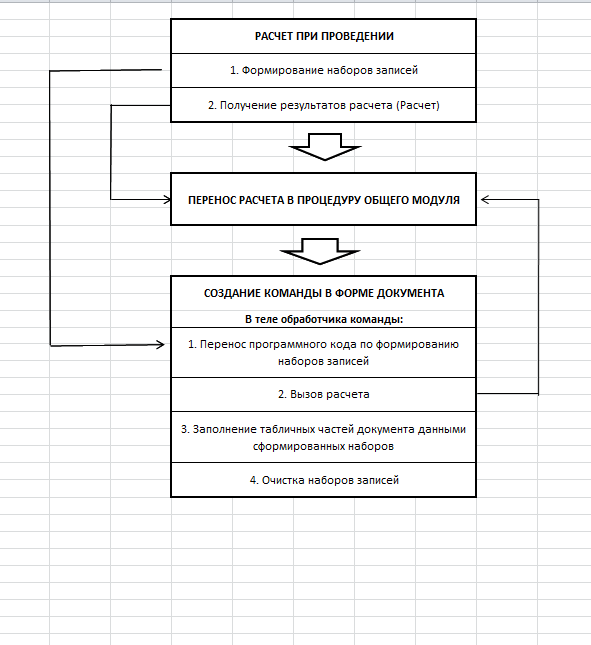
Изменим состав колонок в форме документа, проверим, что структура ТЧ соответствует структуре регистров. Т.к. расчет может быть произведен несколько раз, то нужно:

1) Очищать движения перед заполнением ТЧ. Загрузили движения через метод Выгрузить() для выгрузки движений и загрузить через методы движений.

Затем очистим наборы движений и запишем пустые наборы

В форме сначала были пустые движения, затем их заполнили расчетными значениями.

Движения нужно очищать для следующего проведения расчета . необходимо снова реализовать проведение, т.к. весь код обработчика вынесли в команду. конструктором формируем набор записей. проверим корректность на практике.



### 26.3) Отчет:

Расчет зарплаты по табелю: Отчет

Разберем отчет. Отчет реализован объединением, т.к. нужны данные обеих таблиц. Запрос к реальным таблицам, итоговая колонка - рассчитывается(вычисляемое поле)Но по начальному значению оклада суммирования быть не должно, т.е. расчет только по сотруднику. Отчет строго за месяц. Из вирт. табл. возьмем нач. и кон. периоды, сделаем рассчитываемыми. Если отчет - не кросс - таблица, заголовок создаем макетом. Чтобы вывести заголовок добавляем группировку (Детальные записи), Др. настройки -> Вариант использования группировки -> ДопИнформация

## 27)Перетекающий период:

Разбитие на периоды на границе месяца при еженедельных начислениях, прямо в запросе. Часто встречаются задачи с еженедельными начислениями, одна из особенностей в этом случае - переход через месяц в пределах одной недели. В этом случае нужно получить две записи регистра расчета, соответствуют одной записи документа.

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.ПериодРегистрации,

| ВложенныйЗапрос.Сотрудник КАК Сотрудник,

| ВложенныйЗапрос.Подразделение КАК Подразделение,

| ВложенныйЗапрос.ВидРасчета КАК ВидРасчета,

| ВложенныйЗапрос.Размер,

| ВложенныйЗапрос.ПериодДействияНачало КАК ПериодДействияНачало,

| ВложенныйЗапрос.ПериодДействияКонец

|ИЗ

| (ВЫБРАТЬ

| НачислениеЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации КАК ПериодРегистрации,

| НачислениеЗарплатыОсновныеНачисления.Сотрудник КАК Сотрудник,

| НачислениеЗарплатыОсновныеНачисления.Подразделение КАК Подразделение,

| НачислениеЗарплатыОсновныеНачисления.ВидРасчета КАК ВидРасчета,

| НачислениеЗарплатыОсновныеНачисления.Размер КАК Размер,

| НачислениеЗарплатыОсновныеНачисления.ДатаНачала КАК ПериодДействияНачало,

| ВЫБОР

| КОГДА НАЧАЛОПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаНачала, МЕСЯЦ) <> НАЧАЛОПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаОкончания, МЕСЯЦ)

| ТОГДА КОНЕЦПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаНачала, МЕСЯЦ)

| ИНАЧЕ КОНЕЦПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаОкончания, ДЕНЬ)

| КОНЕЦ КАК ПериодДействияКонец,

| НачислениеЗарплатыОсновныеНачисления.НомерСтроки КАК НомерСтроки

| ИЗ

| Документ.НачислениеЗарплаты.ОсновныеНачисления КАК НачислениеЗарплатыОсновныеНачисления

| ГДЕ

| НачислениеЗарплатыОсновныеНачисления.Ссылка = &Ссылка

| ОБЪЕДИНИТЬ ВСЕ

| ВЫБРАТЬ

| НачислениеЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,

| НачислениеЗарплатыОсновныеНачисления.Сотрудник,

| НачислениеЗарплатыОсновныеНачисления.Подразделение,

| НачислениеЗарплатыОсновныеНачисления.ВидРасчета,

| НачислениеЗарплатыОсновныеНачисления.Размер,

| НАЧАЛОПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаОкончания, МЕСЯЦ),

| КОНЕЦПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаОкончания, ДЕНЬ),

| НачислениеЗарплатыОсновныеНачисления.НомерСтроки

| ИЗ

| Документ.НачислениеЗарплаты.ОсновныеНачисления КАК НачислениеЗарплатыОсновныеНачисления

| ГДЕ

| НачислениеЗарплатыОсновныеНачисления.Ссылка = &Ссылка

| И НАЧАЛОПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаНачала, МЕСЯЦ) <>

|НАЧАЛОПЕРИОДА(НачислениеЗарплатыОсновныеНачисления.ДатаОкончания, МЕСЯЦ)) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| ВложенныйЗапрос.НомерСтроки,

| ПериодДействияНачало";

Пока Выборка.Следующий() Цикл

НоваяЗапись = Движения.ОсновныеНачисления.Добавить();

ЗаполнитьЗначенияСвойств(НоваяЗапись, Выборка);

КонецЦикла;

## 28) МногократноеИзменениеОклада:

Разбитие оклада при многократном изменении его значения внутри периода расчета. Вариант получения разбивки прямо в запросе. Последовательность действий при построении запроса приведена на рисунке 3 и рисунке 4

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Запрос.УстановитьПараметр("ДатаНачала", НачалоМесяца(ПериодРегистрации));

Запрос.УстановитьПараметр("ДатаОкончания", КонецМесяца(ПериодРегистрации));

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.Период,

| ВложенныйЗапрос.Сотрудник,

| ВложенныйЗапрос.Подразделение,

| ВложенныйЗапрос.Оклад

|ПОМЕСТИТЬ втИзмененияОклада

|ИЗ

| (ВЫБРАТЬ

| СведенияОСотрудникахСрезПоследних.Период КАК Период,

| СведенияОСотрудникахСрезПоследних.Сотрудник КАК Сотрудник,

| СведенияОСотрудникахСрезПоследних.Подразделение КАК Подразделение,

| СведенияОСотрудникахСрезПоследних.Оклад КАК Оклад

| ИЗ

| РегистрСведений.СведенияОСотрудниках.СрезПоследних(&ДатаНачала, ) КАК СведенияОСотрудникахСрезПоследних

|

| ОБЪЕДИНИТЬ ВСЕ

|

| ВЫБРАТЬ

| СведенияОСотрудниках.Период,

| СведенияОСотрудниках.Сотрудник,

| СведенияОСотрудниках.Подразделение,

| СведенияОСотрудниках.Оклад

| ИЗ

| РегистрСведений.СведенияОСотрудниках КАК СведенияОСотрудниках

| ГДЕ

| СведенияОСотрудниках.Период МЕЖДУ &ДатаНачала И &ДатаОкончания) КАК ВложенныйЗапрос

|;

|

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

|ВЫБРАТЬ

| ВложенныйЗапрос.ПериодРегистрации,

| ВложенныйЗапрос.Сотрудник КАК Сотрудник,

| ВложенныйЗапрос.Подразделение КАК Подразделение,

| ВложенныйЗапрос.ВидРасчета,

| ВложенныйЗапрос.Оклад,

| ВЫБОР

| КОГДА ВложенныйЗапрос.Период1 ЕСТЬ NULL

| ИЛИ ВложенныйЗапрос.Период1 < ВложенныйЗапрос.ДатаНачала

| ТОГДА ВложенныйЗапрос.ДатаНачала

| ИНАЧЕ ВложенныйЗапрос.Период1

| КОНЕЦ КАК ПериодДействияНачало,

| ВЫБОР

| КОГДА ВложенныйЗапрос.Период2 ЕСТЬ NULL

| ТОГДА КОНЕЦПЕРИОДА(ВложенныйЗапрос.ДатаОкончания, ДЕНЬ)

| ИНАЧЕ ДОБАВИТЬКДАТЕ(ВложенныйЗапрос.Период2, СЕКУНДА, -1)

| КОНЕЦ КАК ПериодДействияКонец

|ИЗ

| (ВЫБРАТЬ

| НачислениеЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации КАК ПериодРегистрации,

| НачислениеЗарплатыОсновныеНачисления.Сотрудник КАК Сотрудник,

| НачислениеЗарплатыОсновныеНачисления.Подразделение КАК Подразделение,

| НачислениеЗарплатыОсновныеНачисления.ВидРасчета КАК ВидРасчета,

| втИзмененияОклада1.Оклад КАК Оклад,

| НачислениеЗарплатыОсновныеНачисления.ДатаНачала КАК ДатаНачала,

| НачислениеЗарплатыОсновныеНачисления.ДатаОкончания КАК ДатаОкончания,

| втИзмененияОклада1.Период КАК Период1,

| МИНИМУМ(втИзмененияОклада2.Период) КАК Период2

| ИЗ

| Документ.НачислениеЗарплаты.ОсновныеНачисления КАК НачислениеЗарплатыОсновныеНачисления

| ЛЕВОЕ СОЕДИНЕНИЕ втИзмененияОклада КАК втИзмененияОклада1

| ЛЕВОЕ СОЕДИНЕНИЕ втИзмененияОклада КАК втИзмененияОклада2

| ПО втИзмененияОклада1.Сотрудник = втИзмененияОклада2.Сотрудник

| И втИзмененияОклада1.Подразделение = втИзмененияОклада2.Подразделение

| И втИзмененияОклада1.Период < втИзмененияОклада2.Период

| ПО НачислениеЗарплатыОсновныеНачисления.Сотрудник = втИзмененияОклада1.Сотрудник

| И НачислениеЗарплатыОсновныеНачисления.Подразделение = втИзмененияОклада1.Подразделение

| И (НачислениеЗарплатыОсновныеНачисления.ВидРасчета = ЗНАЧЕНИЕ(ПланВидовРасчета.Основныеначисления.Оклад))

| ГДЕ

| НачислениеЗарплатыОсновныеНачисления.Ссылка = &Ссылка

|

| СГРУППИРОВАТЬ ПО

| НачислениеЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,

| НачислениеЗарплатыОсновныеНачисления.Сотрудник,

| НачислениеЗарплатыОсновныеНачисления.Подразделение,

| НачислениеЗарплатыОсновныеНачисления.ВидРасчета,

| втИзмененияОклада1.Оклад,

| НачислениеЗарплатыОсновныеНачисления.ДатаНачала,

| НачислениеЗарплатыОсновныеНачисления.ДатаОкончания,

| втИзмененияОклада1.Период) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| Сотрудник,

| Подразделение,

| ПериодДействияНачало";

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

Пока Выборка.Следующий() Цикл

НоваяЗапись = Движения.ОсновныеНачисления.Добавить();

ЗаполнитьЗначенияСвойств(НоваяЗапись, Выборка);

КонецЦикла;

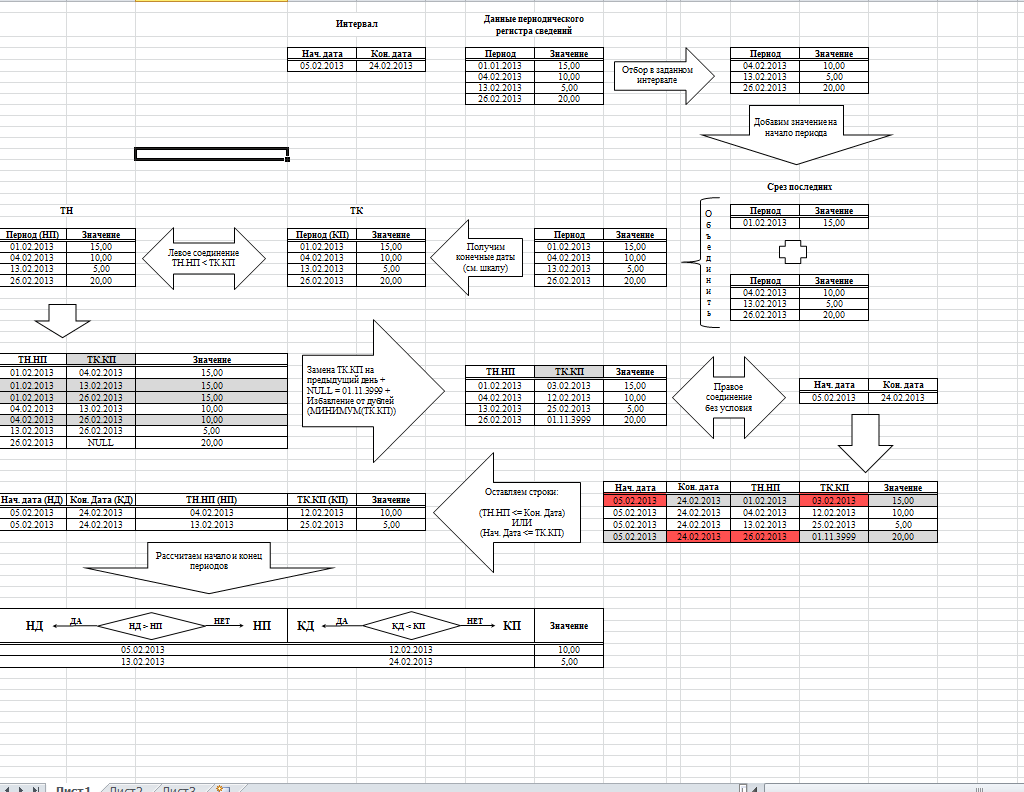


Рисунок 3.Схема построения запроса на получение размера оклада при его многократном изменении.

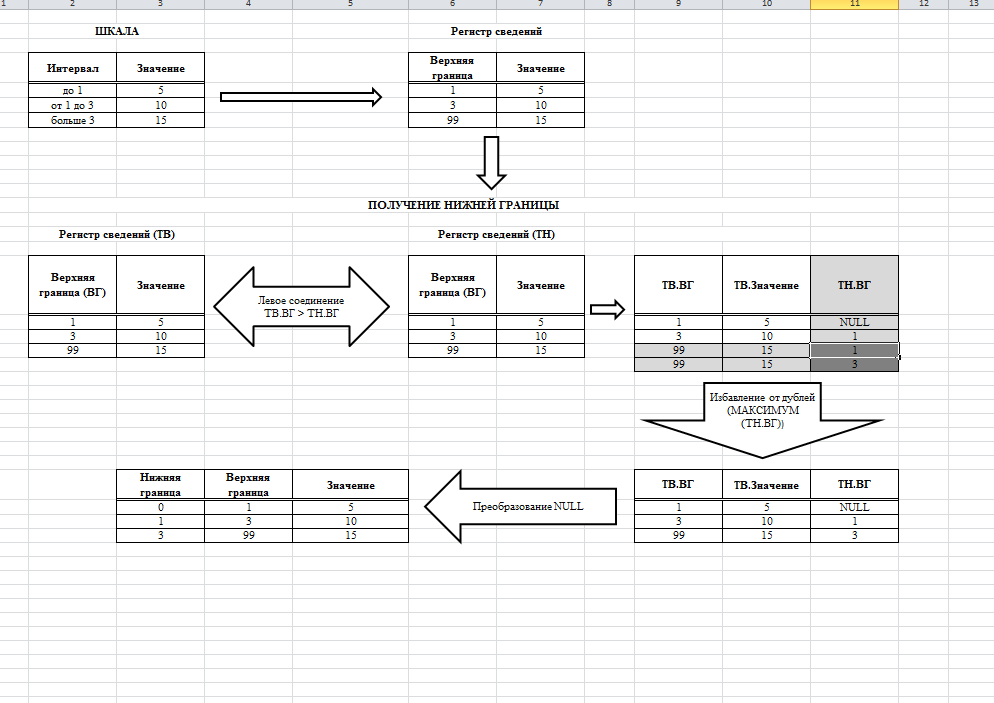


Рисунок 4.Построение «Шкалы»