

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

О.В. Дейнеко

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ОТЕЧЕСТВЕННОГО ПРОИЗВОДСТВА

Учебно-методическое пособие
для лабораторных и практических работ

Ростов-на-Дону
2019

УДК 004.4(07) + 06

Рецензент – кандидат технических наук, доцент В.В. Ильичева

Дейнеко, О.В.

Программное обеспечение отечественного производства: учебно-методическое пособие для лабораторных и практических работ / О.В. Дейнеко; ФГБОУ ВО РГУПС. – Ростов н/Д, 2019. – 40 с. – Библиогр.: с. 40.

Содержатся теоретические сведения, необходимые для выполнения лабораторных и практических работ по дисциплине «Программное обеспечение отечественного производства». Приведены варианты заданий для лабораторных и практических занятий, разобраны примеры решений таких заданий.

Предназначено для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.02 «Информационные системы и технологии».

Одобрено к изданию кафедрой «Вычислительная техника и автоматизированные системы управления».

Учебное издание

Дейнеко Олег Владимирович

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОТЕЧЕСТВЕННОГО ПРОИЗВОДСТВА

Печатается в авторской редакции
Технический редактор Т.И. Исаева

Подписано в печать 22.08.19. Формат 60×84/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 2,32.
Тираж экз. Изд. № 5063. Заказ .

Редакционно-издательский центр ФГБОУ ВО РГУПС.

Адрес университета: 344038, г. Ростов н/Д, пл. Ростовского Стрелкового Полка
Народного Ополчения, д. 2.

© Дейнеко О.В., 2019
© ФГБОУ ВО РГУПС, 2019

СОДЕРЖАНИЕ

Введение	4
Лабораторная (практическая) работа 1	5
Лабораторная (практическая) работа 2	9
Лабораторная (практическая) работа 3	13
Лабораторная (практическая) работа 4	14
Лабораторная (практическая) работа 5	17
Лабораторная (практическая) работа 6	22
Лабораторная (практическая) работа 7	27
Лабораторная (практическая) работа 8	30
Библиографический список.....	40

ВВЕДЕНИЕ

Дисциплина «Программное обеспечение отечественного производства» является продолжением дисциплины «Программирование в 1С». В рамках предшествующей дисциплины студенты изучили устройство технологической платформы «1С:Предприятие 8.3», основные объекты метаданных, приемы работы со встроенным языком программирования и основные синтаксические конструкции, а также язык запросов. В рамках данной дисциплины делается упор на отработку практических навыков использования объектов метаданных и возможностей платформы для построения реальных учетных систем, включая бухгалтерский и складской учет. Основное внимание будет уделено регистрам сведений, регистрам накопления, регистрам бухгалтерии и планам счетов.

Пособие содержит необходимый теоретический материал, указания по выполнению лабораторных и практических работ, примеры выполнения таких работ.

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 1

Тема работы: Регистры накопления.

Цель работы: научиться работать с регистрами накопления.

Теоретическая база

Объект метаданных «Регистр накопления» предназначен для хранения информации в разрезе измерений с использованием ресурсов. Каждая запись регистра накопления представляет из себя строку с определенным разработчиком количеством граф. Графы могут быть заданы измерениями, ресурсами и реквизитами записи регистра.

Измерения – свойства записи регистра, по которой записи проверяются на уникальность. Одновременно в регистре не может существовать двух записей с полностью одинаковым набором значений измерений.

Например, если для регистра накопления «ОстаткиТовараНаСкладе» заданы измерения «Номенклатура» и «Склад», то записи со значениями измерений «Стул», «Центральный склад» и «Стул», «Дополнительный склад» могут существовать в регистре одновременно, а вот две записи с измерениями «Стул», «Центральный склад» – не могут.

Часть измерений для регистров являются стандартными. Регистры накопления не поддерживают прямую запись, данные в такие регистры могут быть внесены только на основании документов-регистраторов. Поэтому первое измерение, которое всегда есть в любом регистре накопления, – «Регистратор». Также все регистры накопления являются периодическими, поэтому второе стандартное измерение – «Период». Третье стандартное измерение – «Номер строки».

Регистры накопления могут быть двух видов: «Остатки» и «Обороты». В случае если для регистра накопления определен вид «Остатки», для каждой записи регистра должно быть определено дополнительное измерение – «Вид движения». Движения для регистра накопления вида «Остатки» могут быть двух видов: «Приход» и «Расход». При определении остатков значения ресурсов записей вида «Расход» вычитаются из значения начального остатка соответствующего ресурса, а значения ресурсов записей вида «Приход» – прибавляются.

Ресурсы – поля регистра, в которые вносится числовая информация для последующего анализа, например расчета остатков или оборотов. Приведенный выше пример регистра с измерениями «Номенклатура» и «Склад» может быть дополнен ресурсами «Количество», «Цена» и «Сумма». Ресурсы не проверяются при вычислении уникальности записей.

Реквизиты – поля регистра, предназначенные для внесения дополнительной информации о записи регистра, данные поля не учитываются при вычислении уникальности строки и не используются для вычисления оборотов и остатков.

Задание: создать в каркасной базе регистр накопления «Остатки деталей» вида остатки. Измерения регистра – склад, материал. Ресурсы – количество. Со-

здать в каркасной базе регистр накопления «ОборотПредприятия» с видом «Обороты» Измерение регистра – номенклатура. Ресурсы – сумма. Настроить регистраторы и движения для обоих регистров.

Порядок выполнения работы

- 1 В дереве конфигурации выбрать ветвь «Регистры накопления», вызвать контекстное меню на нажать «Добавить».
- 2 В поле «Наименование» ввести «ОстаткиДеталей», синоним будет сгенерирован автоматически.
- 3 Выбрать вид регистра «Остатки»
- 4 Добавить регистр накопления в подсистемы «Оказание услуг», «Учет материалов» и «Бухгалтерия»
- 5 На вкладке «Регистраторы» указать в качестве регистраторов документы «ПриходнаяНакладная» и «ОказаниеУслуг»
- 6 На вкладке данные добавить измерения:
 - 6.1. «Склад», типа СправочникСсылка.Склады;
 - 6.2. Материал, типа СправочникСсылка.Номенклатура.
- 7 Добавить ресурс «Количество», типа число, неотрицательное, длина – 10, точность – 2.

Итог настройки регистра показан на рис. 1.

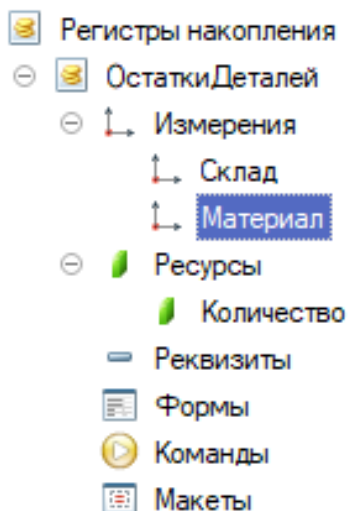


Рисунок 1 – Регистр накопления «ОстаткиДеталей»

После настройки регистра необходимо настроить процесс записи данных из объектов-регистраторов в регистр. Настройка таких записей производится непосредственно в документах-регистраторах на вкладке «движения» с помощью конструктора движений.

Кнопка «Конструктор движений» (рис. 2) вызовет дополнительную форму (3). В конструкторе необходимо сопоставить реквизиты документа с измерениями и ресурсами регистра в который будет производиться запись.

- 8 Настроим движения для документа «Приходная накладная». Очевидно, что видом движения в этом случае будет «Приход». В поле «Табличная часть»

выберем табличную часть «Материалы» приходной накладной. В правой верхней части Показан список реквизитов документа, в том числе поля табличной части. При нажатии на измерение или ресурс в нижней части реквизиты документа, соответствующие по типу выбранному полю регистра, будут отмечены галочкой. Также можно воспользоваться кнопкой «Заполнить выражения». При нажатии этой кнопки платформа автоматически сопоставит поля регистра и реквизиты документа, совпадающие по типу и названию. Итог настройки движения показан на рис. 3.

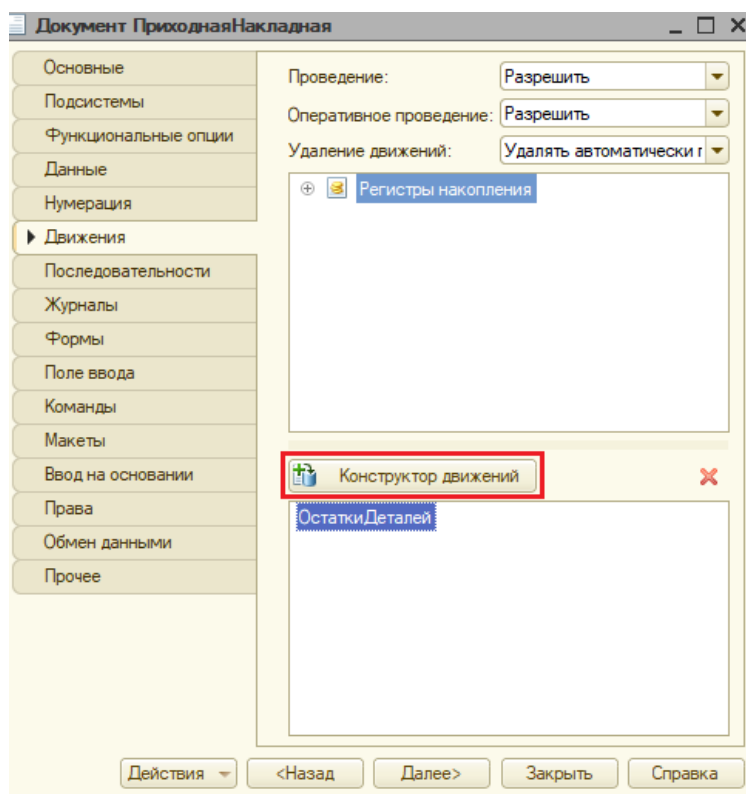


Рисунок 2 – Движения документа

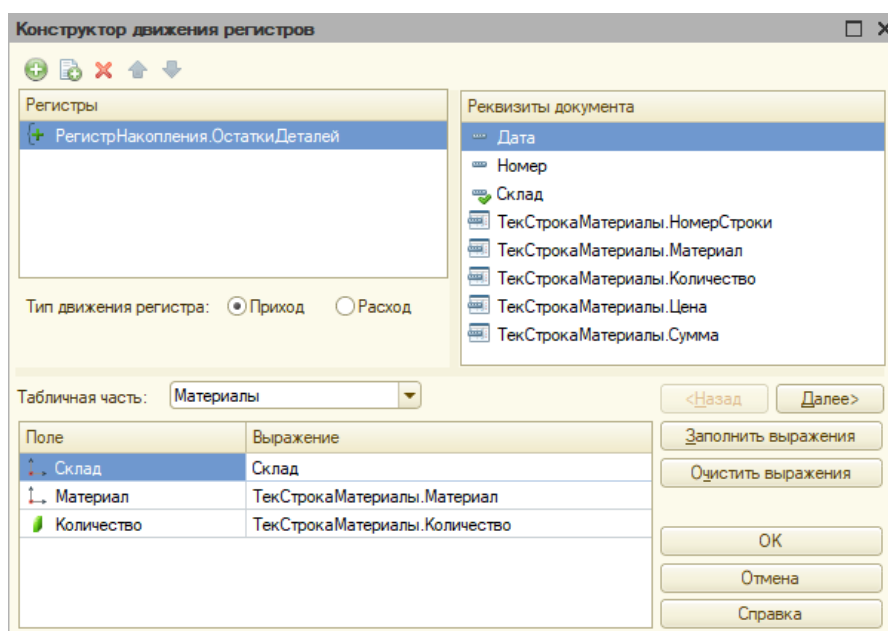


Рисунок 3 – Конструктор движений

После заполнения выражений и нажатия кнопки «ОК» платформа сгенерирует программный код обработки проведения документа и разместит его в модуле объекта, для которого создавалось движение. В нашем случае код будет размещен в модуле объекта документа «Приходная накладная» (рис. 4).

```

Документ ПриходнаяНакладная: Модуль объекта
□ Процедура ОбработкаПроведения(Отказ, Режим)
  //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

  // регистр ОстаткиДеталей Приход
  Движения.ОстаткиДеталей.Записывать = Истина;
  Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.ОстаткиДеталей.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Склад = Склад;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Количество = ТекСтрокаМатериалы.Количество;
  КонечЦикла;

  //}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  КонечПроцедуры
  
```

Рисунок 4 – Программный код, сгенерированный конструктором движений

Как видно из рисунка, программный код обрамляется специальными комментариями, указывающими на то, что код был создан конструктором. Необходимо помнить, что при повторном обращении к конструктору движений все содержимое обработки проведения будет заменено.

9 Теперь создадим расходное движение в документе «Оказание услуг». Для этого повторим действия, совершенные при создании приходного движения в документе «Приходная накладная».

После автоматического создания движения необходимо обратить внимание на следующий нюанс: по логике нашей системы, в документе «Оказание услуг», помимо использованных деталей, указываются также и оказанные услуги. При этом очевидно, что в регистр «ОстаткиМатериалов» записи об услугах попадать не должны. Таким образом, нам необходимо проанализировать реквизит номенклатуры «Услуга» и записывать движение, только если номенклатура не является услугой.

10 Откроем модуль объекта документа «Оказание услуги» и дополним код, сгенерированный конструктором проверкой интересующего нас условия (рис. 5).

```

Документ ОказаниеУслуги: Модуль объекта
□ Процедура ОбработкаПроведения(Отказ, Режим)
  // регистр ОстаткиДеталей Расход
  Движения.ОстаткиДеталей.Записывать = Истина;
  Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если НЕ ТекСтрокаПереченьНоменклатуры.Номенклатура.Услуга Тогда
      Движение = Движения.ОстаткиДеталей.Добавить();
      Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
      Движение.Период = Дата;
      Движение.Склад = Склад;
      Движение.Материал = ТекСтрокаПереченьНоменклатуры.Количество;
    КонечЕсли;
  КонечЦикла;
  КонечПроцедуры
  
```

Рисунок 5 – Программный код обработки проведения документа «Оказание услуги»

- 11 Создадим регистр накопления «ОборотПредприятия» вида «Обороты».
- 12 Действия по созданию регистра аналогичны действиям при создании регистра вида «Остатки» за исключением некоторых нюансов.
- 13 Т. к. оборот относится к деньгам, в регистр будет записываться сумма, а не количество.
- 14 Т. к. обороты считаются нарастающим итогом, в такой регистр вносятся данные только документов, отвечающих за выставление требований клиентам, т. е. в случае нашей базы документ-регистратор для такого регистра – «Оказание услуги»
- 15 Т. к. для оборотов важны суммы, полученные и за детали, и за услуги, донастройка движения, созданного с помощью конструктора движений, не требуется.

Контрольные вопросы

- 1 Для чего предназначены регистры накопления?
- 2 Какие бывают виды регистров накопления?
- 3 Чем отличаются разные виды регистров накопления?
- 4 Для чего нужны измерения, ресурсы, реквизиты?
- 5 Как определить движения для регистра накопления?

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 2

Тема работы: Простой план счетов.

Цель работы: создание простого плана счетов.

Бухгалтерский учет был придуман для приведения различных видов хозяйственной деятельности к единой системе подсчетов, которая бы позволила оценить экономическое состояние предприятия. В качестве единицы измерения для оценки хозяйственной деятельности в бухгалтерском учете используются деньги. Таким образом, вне зависимости от того, какой процесс происходит в реальной хозяйственной деятельности, для бухгалтерского учета имеется денежный эквивалент такой операции. Каждая операция отражается дебетом (приходом) или кредитом (расходом) по каким-либо счетам. Например, приход на склад 11 тонн сахара по цене 3000 рублей за тонну в бухгалтерском учете может быть отражен дебетом по счету «Поступления от поставщиков» на сумму 33 000 рублей и кредитом по счету «Расчеты с поставщиками». Таким образом, для организации бухгалтерского учета на предприятии необходимо продумать структуру счетов, по которым в организации будет происходить движение денежных средств. Такая структура называется «План счетов» и в среде «1С:Предприятие» реализуется одноименным объектом метаданных.

Планом счетов называется совокупность синтетических счетов, предназначенных для группировки информации о хозяйственной деятельности предприятия. Информация, накапливаемая на таких синтетических счетах, позволяет получить полную картину состояния средств предприятия в денежном выражении.

Помимо кода и наименования, механизм работы с планами счетов позволяет создавать набор реквизитов для хранения любой дополнительной информации об элементе плана счетов.

Для каждого плана счетов может быть задано несколько форм просмотра и редактирования.

Наряду с общими свойствами, присущими всем объектам метаданных, планы счетов обладают рядом специфических свойств.

Редактирование плана счетов выполняется в окне редактирования.

Планы счетов в системе «1С:Предприятие» поддерживают многоуровневую иерархию «счет – субсчета». Каждый план счетов может включать неограниченное число счетов первого уровня. К каждому счету может быть открыто также неограниченное количество субсчетов. В свою очередь, каждый субсчет может иметь свои субсчета – и так далее.

На закладке «Данные» формируются реквизиты плана счетов и определяются виды учета (количественный, валютный и т. д.).

Если в Конфигураторе у плана счетов установлено свойство «Автопорядок по коду», то при формировании списка счетов будет отмечено, что сортировка выполняется по полю «Код». В этом случае свойство «Длина порядка» не может быть меньше, чем длина кода.

Если свойство «Автопорядок по коду» установлено, то колонка «Порядок» становится невидимой и недоступной в форме, сформированной по умолчанию. При этом сортировку можно выполнять по колонкам «Код» и «Наименование». Однако в этом случае сортировка по полю «Код» подменяется сортировкой по полю «Порядок».

Смысл свойства «Автопорядок по коду» именно в том, чтобы наличие самого поля «Порядок» было бы скрыто от пользователя. Это свойство обеспечивает упорядочивание по коду для пользователя, хотя на самом деле для упорядочивания используется поле «Порядок». Соответственно поле «Порядок» не показывается по умолчанию пользователю, а разработчик должен обеспечить его программное заполнение.

Если предполагается, что пользователь может просматривать и редактировать это поле, то свойство «Автопорядок по коду» не должно быть установлено.

На закладке «Субконто» выбираются субконто из набора планов видов характеристик и создаются «признаки учета субконто», т. е. список видов учета. Типовым примером служит валютный и количественный учет.

На закладке «Прочее» задаются predeterminedенные счета, управляемые транзакционные блокировки данных по полям, устанавливаются признаки полнотекстового поиска и использования истории данных по объектам данного типа.

Pредeterminedенные счета плана счетов вводятся в отдельном окне, вызываемом по кнопке «Предeterminedенные». Эти счета не могут быть удалены пользователями в режиме «1С:Предприятие».

Для установки режимов обновления predeterminedенных данных в поле «Обновление predeterminedенных данных» следует выбрать требуемый режим:

- «Авто» – фактическое значение вычисляется автоматически. Для главного узла информационной базы – значение будет равно «Обновлять автоматически», для периферийного узла информационной базы будет равно «Не обновлять автоматически»;

- «Обновлять автоматически» – при реструктуризации информационной базы будет выполняться автоматическое создание предопределенных элементов и обновление существующих значений;

- «Не обновлять автоматически» – при реструктуризации информационной базы не будет выполняться автоматическое создание новых предопределенных элементов и обновление их значений.

В свойстве «Поля блокировки данных» устанавливается перечень допустимых полей, по которым может быть установлена управляемая блокировка.

Свойство «Полнотекстовый поиск» позволяет включать («Использовать») или исключать («Не использовать») данные объекта в/из полнотекстового индексирования.

Свойство «История данных» позволяет включать («Использовать») или исключать («Не использовать») данные объекта в/из версионирования данных.

Порядок выполнения работы

1 Добавить в каркасную конфигурацию объект метаданных план счетов, в качестве имени, указать «ПростойПланСчетов».

2 Включить план счетов в подсистему «Бухгалтерия».

3 На вкладке данные (рис. 6) установить длину кода 7 и задать маску вида @.@.@@. Маска такого вида означает, что код счета должен будет содержать любую цифру, точку, любую цифру, точку и две любых цифры. Однако, следует помнить, что маска, заданная в редакторе плана счетов, будет работать только для обычных форм, если планируется использование управляемых форм, настройка должна быть продублирована в палитре свойств плана счетов. Выбрать основное представление в виде кода. Добавить реквизит «ПолноеНаименование» типа «строка» длиной 150 символов для хранения полного наименования счета. Добавить признаки учета «ИспользоватьСумму», «НужноКоличество», «ВестиВалютныйУчет». Признаки учета, по сути, являются реквизитами плана счетов, от обычных реквизитов их отличает то, что все признаки учета могут быть лишь одного типа – булево. Добавить табличную часть «Ответственные» с одним реквизитом «Сотрудник» типа «СправочникСсылка.Сотрудники».

4 На вкладке «Прочее» перейти в форму «Предопределенные» и произвести настройку предопределенных элементов плана счетов в соответствии с рис. 7.

Контрольные вопросы

- 1 Для чего предназначены планы счетов?
- 2 Как настроить маску плана счетов для обычного и для управляемого приложения?
- 3 Для чего используются признаки учета?
- 4 Сколько уровней вложенности поддерживают счета?

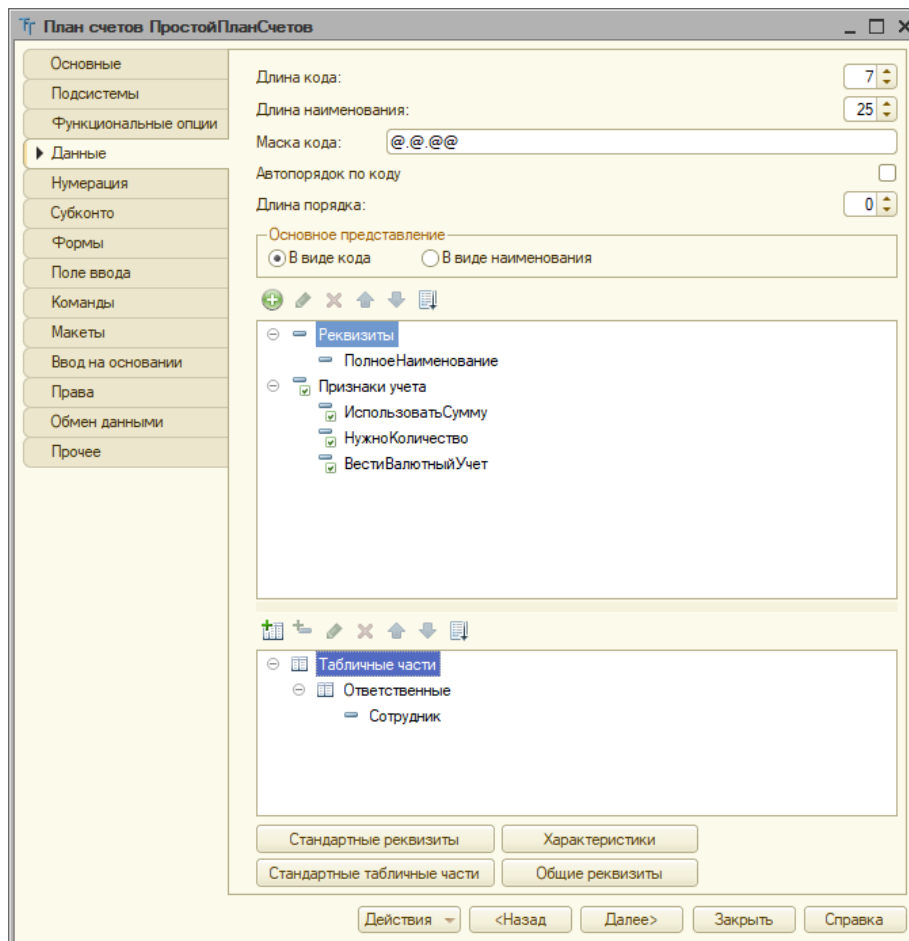


Рисунок 6 – Вкладка «Данные» редактора плана счетов

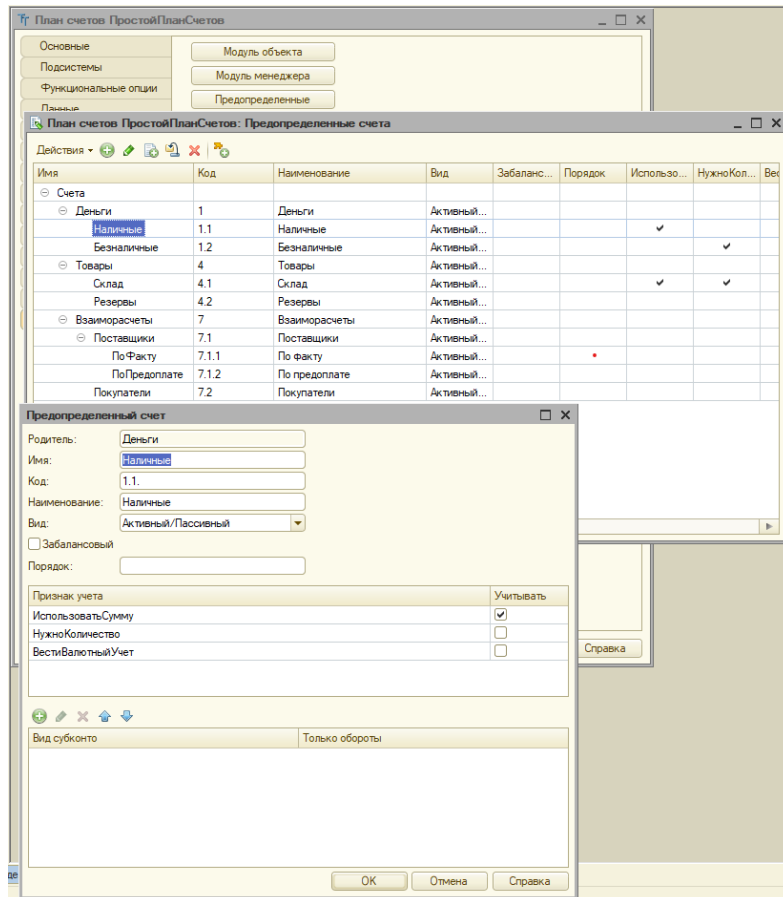


Рисунок 7 – Настройка предопределенных элементов

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 3

Тема работы: простой регистр бухгалтерии.

Цель работы: создание простого регистра бухгалтерии, связанного с созданным ранее планом счетов.

Для отражения в бухгалтерском учете информации о хозяйственных операциях в системе «1С:Предприятие» используются регистры бухгалтерии, описываемые на ветви дерева конфигурации «Регистры бухгалтерии».

По своему виду регистр бухгалтерии напоминает регистр накопления. При его редактировании разрабатывается структура регистра: создаются наборы измерений, ресурсов и реквизитов регистра, если необходимо, создаются экранные и печатные формы просмотра движений регистра.

Отличительной особенностью регистра бухгалтерии является его связь с планом счетов и поддержка механизма двойной записи. Каждая запись регистра содержит обязательные реквизиты «Счет Дт» (счет дебета) и «Счет Кт» (счет кредита) для регистров, поддерживающих корреспонденцию, и реквизит «Счет» для не поддерживающих.

Редактирование регистра бухгалтерии выполняется в окне редактирования.

Наряду с общими свойствами, присущими всем объектам конфигурации, регистры бухгалтерии обладают рядом специфических свойств. На закладке «Основные» выбирается план счетов и указывается, поддерживает ли данный регистр корреспонденцию счетов. Свойство «Длина уточнения периода» используется для дополнительного упорядочивания движений в регистре по оси времени. Свойство может принимать целочисленное значение от «0» до «3». Если установлено значение «0», регистр бухгалтерии не поддерживает уточнение периода.

Если уточнение периода используется, то при сравнении двух движений с равными периодами более ранним по времени будет считаться движение с меньшим значением уточнения периода.

Порядок выполнения работы

1 Создать объект метаданных регистр бухгалтерии. В качестве имени задать «ПростойРегистрБухгалтерии».

2 В качестве используемого плана счетов указать «ПростойПланСчетов».

3 Включить регистр в подсистему «Бухгалтерия»

4 Добавить в регистр следующие измерения: «Организация» типа «СправочникСсылка.Организации», «Валюта» типа «СправочникСсылка.Валюты». Для измерения «Валюта» установить признак учета «ВестиВалютныйУчет».

5 Добавить в регистр следующие ресурсы: «Сумма» типа «ОпределяемыйТип.Деньги», «Количество» типа «Число», «ВалютнаяСумма» типа «ОпределяемыйТип.Деньги».

6 Определить признаки учета для всех ресурсов. «Сумма» – «ИспользоватьСумму», «Количество» – «НужноКоличество», «ВалютнаяСумма» – «ВестиВалютныйУчет». После завершения данных шагов вкладка данные регистра должна иметь вид, показанный на рис. 8.

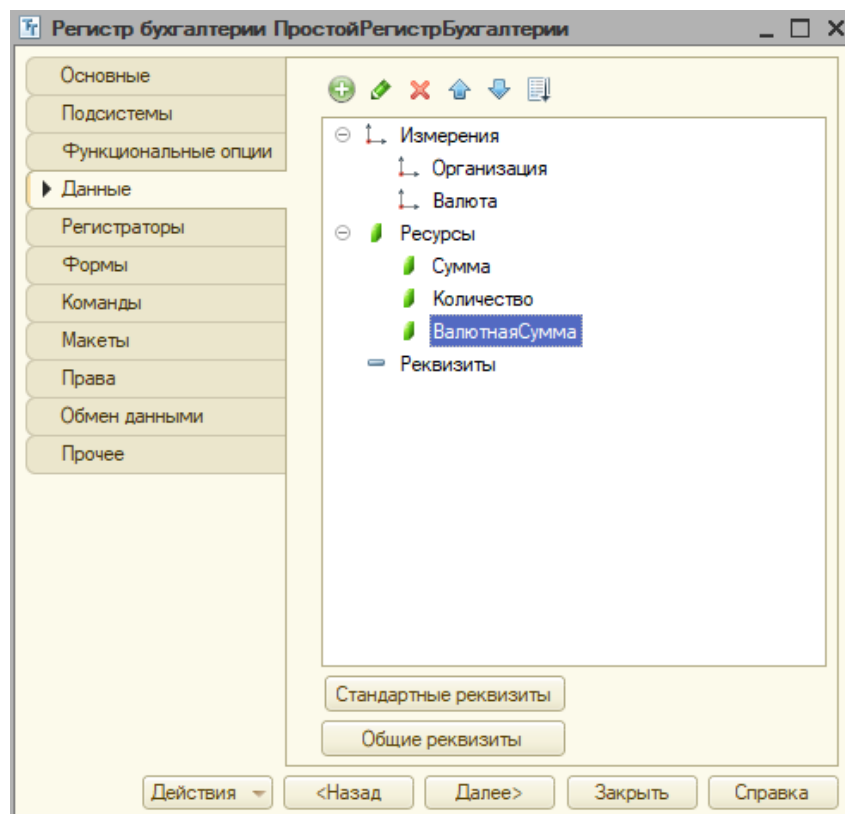


Рисунок 8 – Вкладка данные редактора регистра бухгалтерии

7 Добавить в конфигурацию документ «ПростаяОперация».

8 Указать документ в качестве регистратора для регистра бухгалтерии «ПростойРегистрБухгалтерии».

9 Создать форму документа «ПростаяОперация».

10 Добавить на форму документа движения по регистру «ПростойРегистрБухгалтерии».

11 Запустить конфигурацию в режиме «1С:Предприятие» и создать 10–15 документов «ПростаяОперация» с различным набором движений.

Контрольные вопросы

- 1 Для чего предназначены регистры бухгалтерии?
- 2 Для чего нужны измерения и ресурсы регистра?
- 3 Какие режимы записи поддерживает регистр бухгалтерии?
- 4 Как настраиваются признаки учета для измерений и ресурсов?

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 4

Тема работы: Регистр бухгалтерии с настраиваемыми аналитическими разрезами и поддержкой корреспонденции.

Цель работы: усложнение структуры регистра бухгалтерии с целью ведения учета с настройкой аналитических разрезов учета балансовых признаков учета.

Для любого плана счетов в системе «1С:Предприятие» могут быть определены т. н. «субконто». Субконто, по своей сути, является дополнительным ресурсом для регистра бухгалтерии, с которым связывается план счетов, для которого настраиваются субконто. Таким образом, субконто позволяют получать отчеты по регистрам бухгалтерии в разрезе дополнительных аналитических показателей. Т. к. заранее предусмотреть, в каком разрезе понадобится аналитика, пользователь не может, в системе «1С:Предприятие» предусмотрена возможность настройки аналитических разрезов через объект метаданных «План видов характеристик». В данной лабораторной работе мы создадим план видов характеристик, который позволит пользователю настраивать аналитические разрезы для ведения бухгалтерского учета.

Для каждого плана счетов может быть включена настройка «Корреспонденция», отвечающая за управление принципом двойной записи в регистре бухгалтерии. По факту, если корреспонденция отсутствует, каждая запись имеет только одно поле – «Счет», а если корреспонденция включена, то каждая запись имеет обязательные поля «СчетКт» и «СчетДт» для фиксации того, по какому счету происходит кредит, а по какому – дебет.

Также для каждого измерения может быть осуществлена настройка «Балансовый». Включение этой настройки для измерения означает, что в разрезе этого измерения будет вестись учет для каждой записи, в противном случае необходимо будет указывать своё значение для дебета и кредита. Пример – измерение «Организация» почти всегда балансовое, т. к. используется во всех проводках. Однако измерение «Валюта» требуется не во всех записях, поэтому оно не балансовое.

Порядок выполнения работы

- 1 Создать новый объект метаданных «План видов характеристик».
- 2 Задать имя плана «ВидыАналитическихРазрезов».
- 3 Добавить в качестве типов значения характеристик следующие типы данных: «СправочникСсылка.ЮридическиеЛица», «СправочникСсылка.Сотрудники», «ДокументСсылка.ЗаказПокупателя», «СправочникСсылка.Склады», «СправочникСсылка.Контрагенты», «СправочникСсылка.ФизическиеЛица», «СправочникСсылка.Номенклатура». В данном случае мы ограничиваем возможности пользователя по созданию аналитических разрезов для регистра бухгалтерии, с которым будет связан данный план видов характеристик. Однако, если в таких ограничениях нет необходимости, в качестве типа значения характеристик может быть указан тип «ЛюбаяСсылка».

- 4 В режиме «1С:Предприятие» добавить следующие элементы (через тире указан тип данных): Заказы покупателей – Заказ покупателя, Контрагенты – Контрагенты, Налоговые агенты – Физические лица, Юридические лица, Номенклатура – номенклатура, Склады – Склады, Сотрудники – Сотрудники. После добавления необходимых данных справочник в режиме «1С:Предприятие» должен иметь вид, показанный на рис. 9.

Код	Наименование	Тип значения
000000001	Контрагенты	Контрагенты
000000002	Номенклатура	Номенклатура
000000003	Заказы покупателей	Заказ покупателя
000000004	Склады	Склады
000000005	Налоговые агенты	Юридические лица, Физические лица
000000006	Сотрудники	Сотрудники

Рисунок 9 – Заполненный справочник «Виды аналитических разрезов»

5 В настройках плана счетов «ПростойПланСчетов», на вкладке «Субконто», выбрать в качестве видов субконто план видов характеристик «Виды аналитических разрезов»

6 Дополнить форму документа простая операция появившимися элементами субконто. После выполнения данного шага настройки форму документа «Простая операция» должны иметь вид, представленный на рис. 10. Несмотря на то что изменения вносились только в план счетов, в регистре бухгалтерии, связанном с этим планом счетов, автоматически появились необходимые графы для учета всех добавленных субконто.

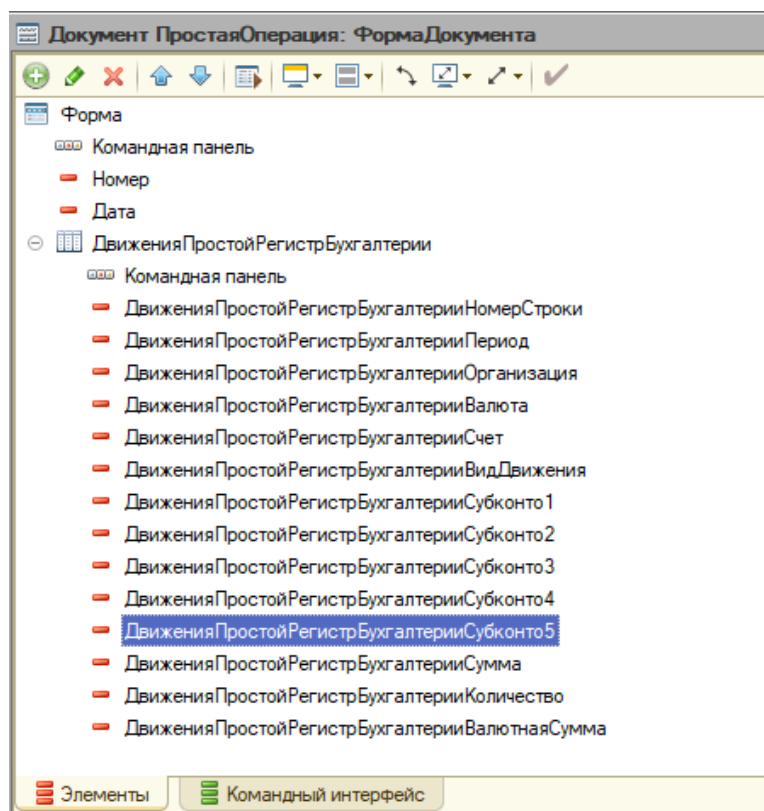


Рисунок 10 – Настройки формы документа «Простая операция»

7 В плане счетов произвести настройку с учетом использованием настраиваемого субконто, настройки счетов производятся в свободной форме подобно настройке счета, показанной на рис. 11.

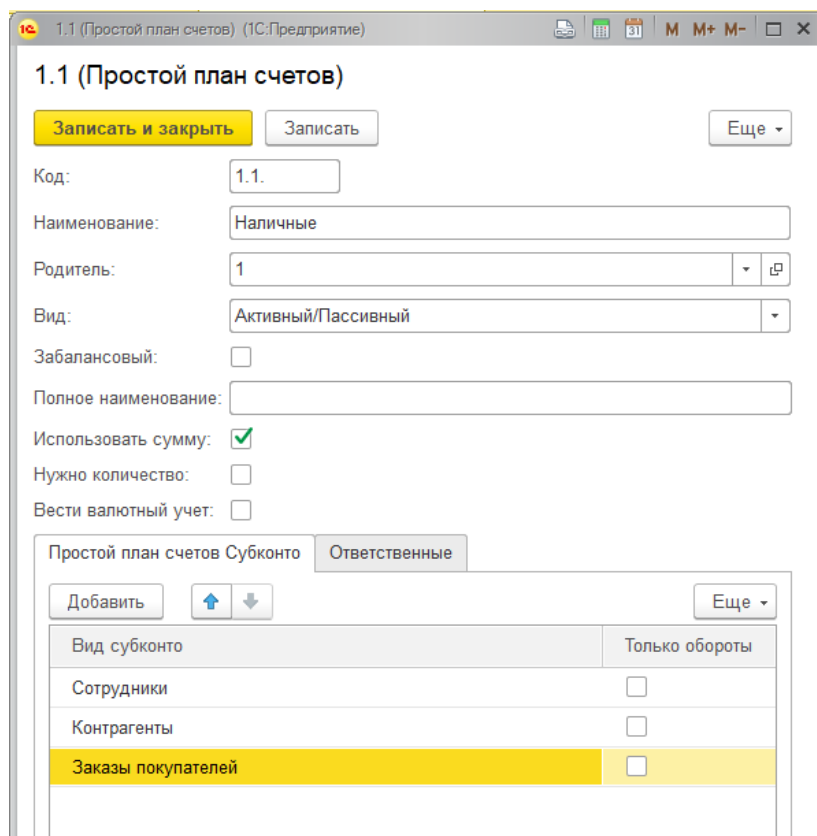


Рисунок 11 – Пример настройки счета

8 В настройках регистра бухгалтерии отметить измерение организация как балансовое.

9 Перезаполнить документы «Простая операция» с учетом новых настроек и заново провести эти документы.

Контрольные вопросы

- 1 Для чего необходимо субконто?
- 2 Какой объект метаданных используется для предоставления пользователю возможности использования субконто?
- 3 Где настраивается сопоставления плана счетов с настраиваемыми видами субконто?
- 4 Отражаются ли субконто в регистрах бухгалтерии?
- 5 Для чего служит флаг «Корреспонденция» в настройках регистра бухгалтерии?
- 6 Для чего служит флаг «балансовый» в настройках измерений регистра бухгалтерии?

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 5

Тема работы: Создание оборотно-сальдовой ведомости.

Цель работы: научиться формировать отчеты по регистрам бухгалтерии.

Отчет «Оборотно-сальдовая ведомость» представляет собой таблицу, в строках которой перечислены все имеющиеся в плане счетов счета, а в колонках

– начальное сальдо, оборот и конечное сальдо по дебету и кредиту каждого счета. Сальдо в бухгалтерском учете – остаток по бухгалтерскому счету, разность между суммой записей по дебету и кредиту счетов.

– Дебетовое сальдо (дебет больше кредита) отражает состояние данного вида хозяйственных средств на определенную дату и показывается в активе баланса.

– Кредитовое сальдо (кредит больше дебета) отражает состояние источников хозяйственных средств и показывается в пассиве.

Если счет не имеет остатка (сальдо равно нулю), то такой счет называется закрытым. В бухгалтерском учете некоторые счета могут одновременно иметь и дебетовое, и кредитовое сальдо.

Порядок выполнения работы

1 В режиме Конфигуратор добавьте новый объект «Отчет» с именем «ОборотноСальдоваяВедомость». Создайте новую схему компоновки данных и добавьте Набор данных – запрос. Откройте конструктор запроса. Для построения такого отчета понадобятся две исходные таблицы: объектная (ссылочная) таблица плана счетов «ПростойПланСчетов» и виртуальная таблица регистра бухгалтерии «ПростойРегистрБухгалтерии.ОстаткиИОбороты».

2 Из таблицы «ПростойПланСчетов» выберем поля «Код» и «Наименование», из таблицы «ПростойРегистрБухгалтерии.ОстаткиИОбороты» поля:

- СуммаНачальныйОстатокДт,
- СуммаНачальныйОстатокКт,
- СуммаОборотДт,
- СуммаОборотКт,
- СуммаКонечныйОстатокДт,
- СуммаКонечныйОстатокКт.

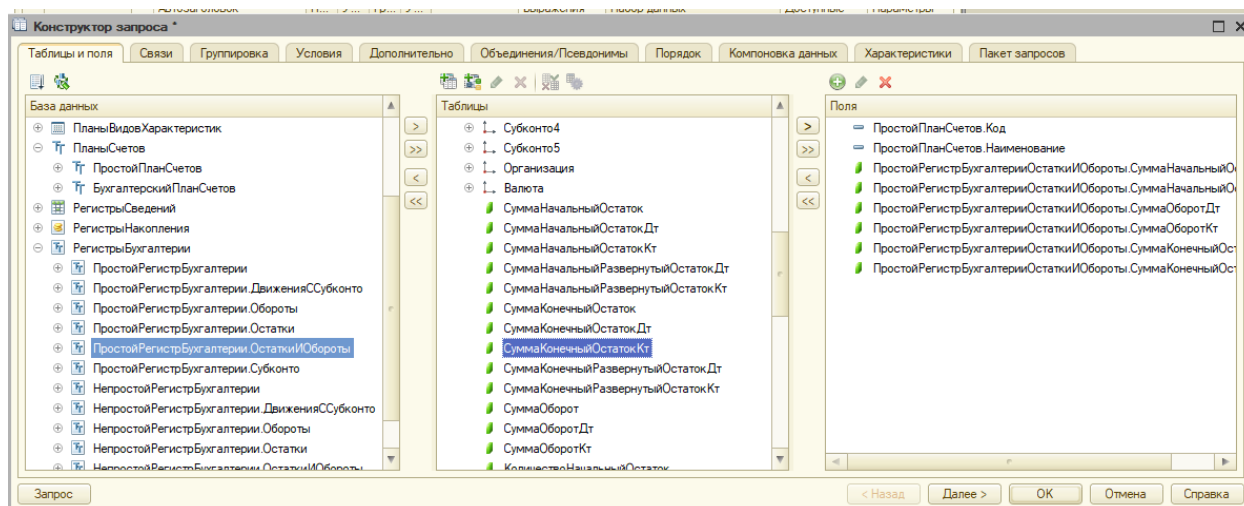


Рисунок 12 – Таблицы и поля запроса

3 Перейдите на закладку «Связи» и укажите, что из таблицы «ПростойПланСчетов» мы будем выбирать все записи, а из таблицы регистра – только те, которые соответствуют условию связи.

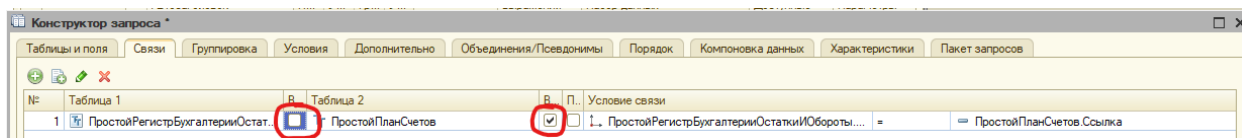


Рисунок 13 – Связи запроса

4 На вкладке Объединения/Псевдонимы задайте псевдонимы полей регистра: СальдоНачДт, СальдоНачКт, ОборотДт, ОборотКт, СальдоКонДт, СальдоКонКт.

Имя поля	Запрос 1
Код	ПростойПланСчетов.Код
Наименование	ПростойПланСчетов.Наименование
СальдоНачДт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаНачальныйОстатокДт
СальдоНачКт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаНачальныйОстатокКт
СуммаОборотДт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаОборотДт
СуммаОборотКт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаОборотКт
СальдоКонДт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаКонечныйОстатокДт
СальдоКонКт	ПростойРегистрБухгалтерииОстаткиИОбороты.СуммаКонечныйОстатокКт

Рисунок 14 – Объединения/Псевдонимы запроса

5 На вкладке Порядок укажите, что результат запроса должен быть отсортирован по возрастанию поля «Код». Нажмите ОК.

6 Роли полей остатка. Теперь, чтобы схема компоновки могла отобразить общие итоги по полям бухгалтерских остатков, внесем небольшие изменения в роли, которые она автоматически определила для полей остатка. Для этих полей система определила бухгалтерский тип – Дебет и Кредит. Поэтому, когда в нашей оборотно-сальдовой ведомости будет рассчитываться общий итог по этим полям, мы получим значение 0, т. к. сумма по дебету будет равна сумме по кредиту, только с обратным знаком. Для того чтобы избежать такой ситуации, в ролях этих полей мы уберем указание бухгалтерского типа и изменим имена групп полей.

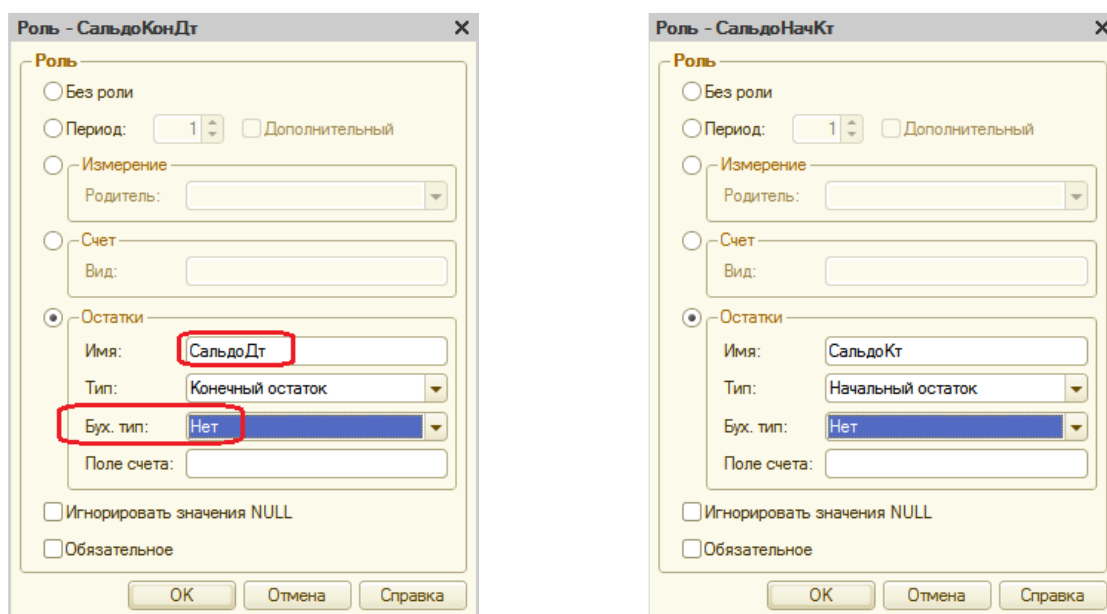


Рисунок 15 – Настройка ролей

В этом случае система компоновки будет воспринимать эти поля как обычные поля остатков. Дважды щелкните на записи в поле «Роль» для ее редактирования и нажмите кнопку выбора. Для полей «СальдоНачДт» и «СальдоКонДт» задайте имя – «СальдоДт». Для полей «СальдоНачКт» и «СальдоКонКт» – имя «СальдоКт». Для всех четырех полей установите «Бухгалтерский тип» в значение «Нет».

В итоге получим:

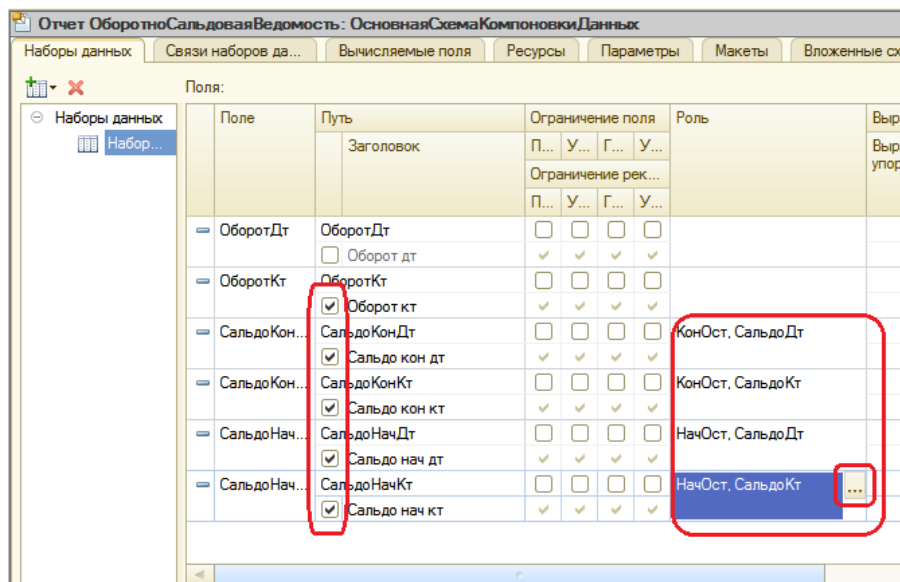


Рисунок 16 – Итог настройки ролей

7 Ресурсы. Перейдите на закладку Ресурсы и добавьте все доступные ресурсы кнопкой >>.

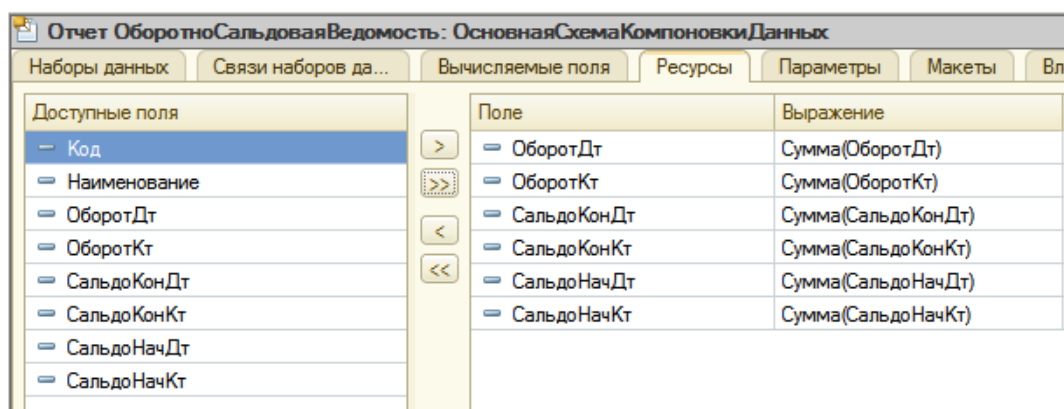


Рисунок 17 – Ресурсы

8 Параметры. Бухгалтерские отчеты формируются, как правило, для определенного периода: месяц, квартал, год. Поэтому на примере нашего отчета продемонстрируем использование стандартного периода для указания периода отчета. На закладке Параметры добавим параметр с именем «Период» типа СтандартныйПериод, а для параметров НачалоПериода и КонецПериода укажем Выражение &Период.ДатаНачала и &Период.ДатаОкончания соответственно для расчета и запретим их редактирование пользователем.

Имя	Заголовок	Тип	Досту...	Д...	Значение	Выражение	Па...	В...	О...
НачалоПериода	Начало периода	Дата		<input type="checkbox"/>		&Период.ДатаНачала	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
КонецПериода	Конец периода	Дата		<input type="checkbox"/>		&Период.ДатаОкончания	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Период	Период	СтандартныйПериод					<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 18 – Параметры

9 Перейдите на закладку «Настройки» и создайте структуру отчета. Добавьте группировку, содержащую детальные записи. Затем на закладке Выбранные поля выберем все поля для вывода в отчет и разместим их в следующем порядке, показанном на рис. 19.

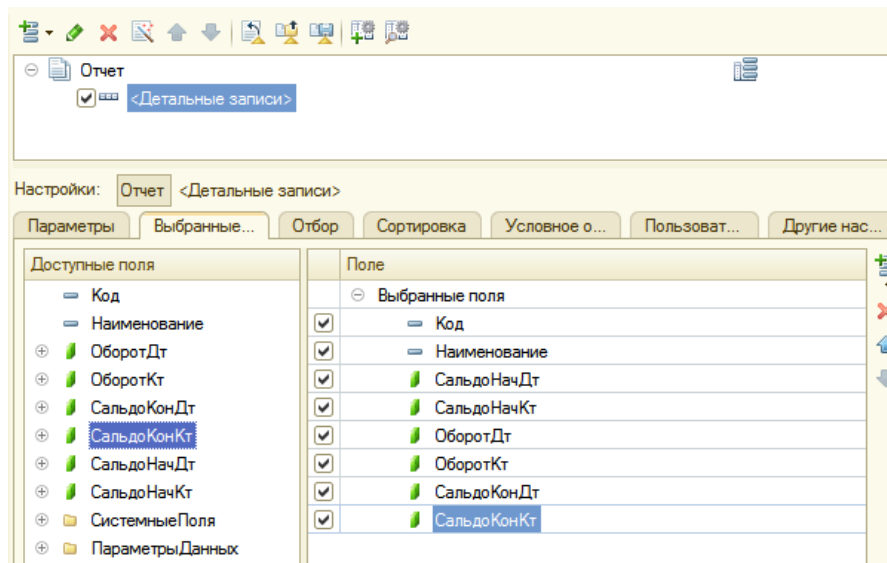


Рисунок 19 – Настройка полей отчета

10 На закладке «Другие настройки» укажем заголовок отчета – «Оборотно-сальдовая ведомость». Для параметра «Расположение общих итогов по вертикали» укажем значение «Начало и конец».

11 На закладке «Параметры» выберем для параметра «Период» значение из списка стандартных периодов – «Этот месяц». Тем самым мы обеспечим указание текущего месяца при открытии формы отчета с динамическим изменением даты.

12 Нажмите кнопку «Свойства элемента» пользовательских настроек, укажите, что параметр «Период» будет включен в состав быстрых пользовательских настроек.

13 В заключение определим, что отчет будет отображаться в подсистеме Бухгалтерия.

14 Запустите отладку, откройте в разделе «Бухгалтерия» отчет и нажмите «Сформировать». Мы видим, что стандартный период отчета задан по умолчанию – «Этот месяц».

Контрольные вопросы

- 1 Как получить данные из регистра бухгалтерии запросом?
- 2 Как создать отчет на основании данных из регистра бухгалтерии с помощью системы компоновки?
- 3 Как задать роли и тип бухгалтерского остатка полям в схеме компоновки данных?
- 4 Как задать стандартный период для выполнения отчета?

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 6

Тема работы: План видов расчета, регистр расчета.

Цель работы: создание плана видов расчета и регистра расчета.

Объект «План видов расчета» предназначен для описания структуры хранения информации о возможных видах расчетов. На основе Плана видов расчета платформа создает в базе данных таблицу, в которой будет храниться информация о том, какие существуют виды расчета и взаимосвязи между ними.

Отличительной особенностью плана видов расчета является то, что пользователь в процессе работы может добавлять новые виды расчета.

План видов расчета имеет следующие свойства:

- Использует период действия – определяет, будут ли в этом плане находиться виды расчета, которые могут быть вытеснены по периоду действия.
- Зависимость от базы – определяет, будут ли в этом плане находиться зависимые по базовому периоду виды расчета.
- Еще одной важной особенностью плана видов расчета является возможность создания предопределенных видов расчета и описания их взаимного влияния. При этом разработчик имеет возможность указать три категории видов расчета, влияющих на предопределенный вид расчета:
- Базовые – их результаты должны быть использованы при перерасчете этого вида расчета.
- Вытесняющие – вытесняют этот вид расчета по периоду действия.
- Ведущие – изменение их результатов должно приводить к необходимости перерасчета этого вида расчета.

Таким образом, с каждым видом расчета будет связано три периода:

- Период действия является «запрашиваемым». Т. е., указывая период действия, мы говорим: «Мы хотели бы, чтобы результат действовал в этом периоде».
- Фактический период – это то, что получилось из периода действия после анализа всех периодов действия расчетов, которые вытесняют наш по периоду действия.
- Базовый период – в котором мы анализируем результаты других расчетов, влияющих на наш по базовому периоду (календарный месяц).

Объект конфигурации «Регистр расчета» предназначен для описания структуры накопления данных, являющихся результатами расчетов. На основе регистра расчета платформа создает в базе данных таблицы, в которых будут накапливаться данные, формируемые различными объектами базы данных.

Отличительные особенности регистра расчета:

- Не предназначен для интерактивного редактирования пользователем. Разработчик может при необходимости предоставить пользователю возможность редактирования, но изначально регистр расчета для этого не предназначен.
- Периодичность. Определяет промежуток времени, к которому будет относиться каждая запись регистра.
- Возможность использования механизмов вытеснения по периоду действия. При этом для каждой записи регистр формирует фактический период действия, который является совокупностью нескольких периодов внутри периода действия.
- Возможность использования механизмов вытеснения по зависимости по базовому периоду. Этот механизм позволяет основывать расчет зависимых (вторичных) записей регистра на данных, полученных в результате расчета первичных записей.
- Зависимость по периоду действия. При анализе базовых записей будут выбираться те, в которых найдено пересечение фактического периода действия и указанного базового периода.
- Зависимость по периоду регистрации. При анализе базовых записей будут выбираться те, которые попадают в указанный базовый период значением своего поля Период регистрации.
- Связь с планом видов расчета. На основе этой связи работают механизмы вытеснения по периоду действия и зависимости по базовому периоду, поскольку в плане видов расчета описано взаимное влияние видов расчета друг на друга.

У регистра расчета могут существовать подчиненные объекты «Перерасчет». Они предназначены для регистрации фактов появления в регистре записей, влияющих на результат расчета уже существующих записей регистра. Перерасчет может иметь несколько измерений, каждое из которых устанавливает связь между измерениями данного регистра расчета и влияющих регистров расчета. В частном случае это может быть один и тот же регистр.

Таблицы перерасчета заполняются автоматически на основании записей регистров расчета, затронутых ведущими видами расчета и на основании записей регистра расчета, для которых изменился фактический период действия.

Регистр расчета может быть связан с графиком времени. Такой график времени должен представлять собой регистр сведений (непериодический, с обязательным измерением типа Дата и ресурсом типа Число), в котором содержится временная схема исходных данных, участвующих в расчетах. Измерениями этого графика могут быть, например, график работы (ссылка на справочник) и дата, а ресурсом – количество рабочих часов в этой дате.

Порядок выполнения работы

1 В режиме Конфигуратор создайте новый объект конфигурации «План видов расчета» с именем «ОсновныеНачисления». Представление списка задайте как Виды расчетов.

2 На закладке Подсистемы укажите – «РасчетЗарплаты».

3 На закладке «Расчет» укажите, что он будет использовать период действия и зависеть от базы по периоду действия. В качестве базового плана видов расчета укажем его самого, поскольку все наши виды расчетов будут храниться в единственном плане видов расчета (рис. 20).

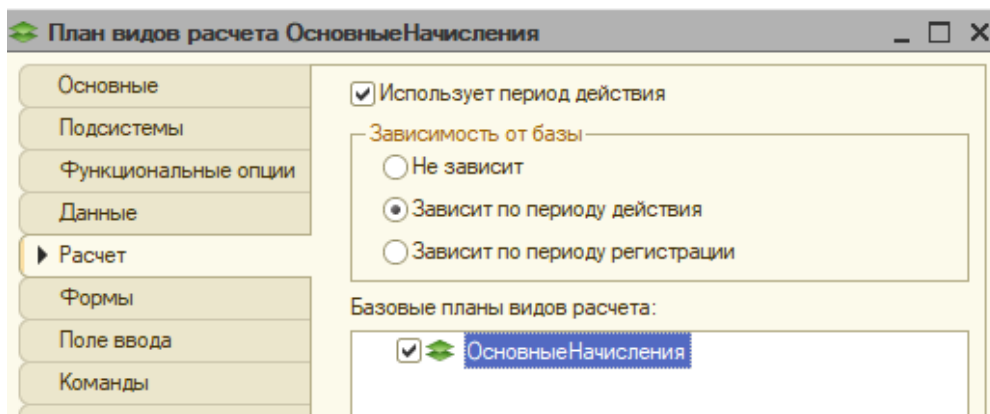


Рисунок 20 – Вкладка «Расчет» плана видов расчета

4 На закладке «Прочее» зададим predetermined types of calculation:

- Невыход – с именем, кодом и наименованием «Невыход»;
- Оклад – с именем, кодом и наименованием «Оклад» и вытесняющим его видом расчета «Невыход»;
- Премия – с именем, кодом и наименованием «Премия», с базовым видом расчета «Оклад» и ведущими видами расчета «Невыход» и «Оклад» (рис. 21).

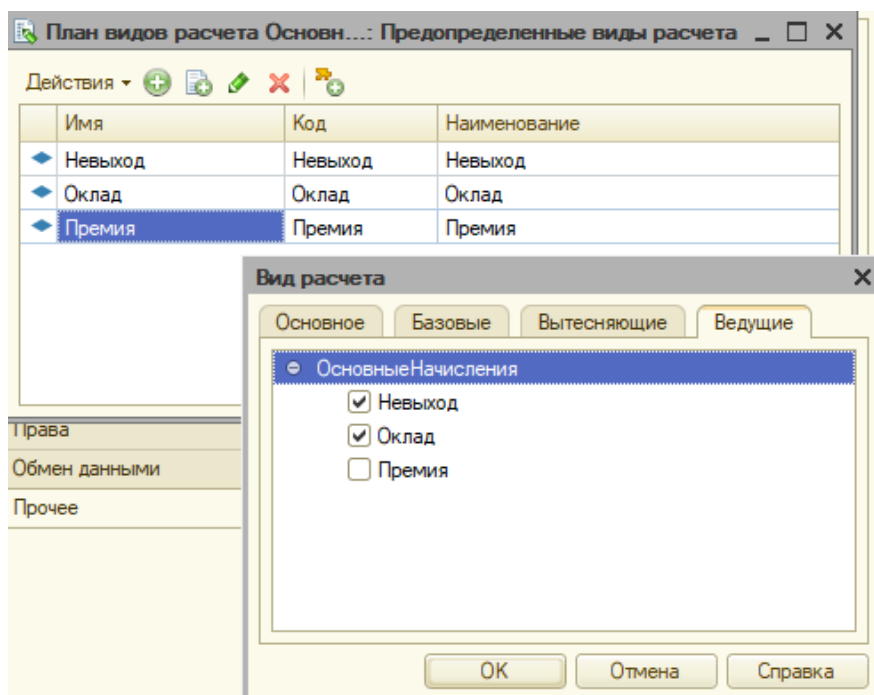
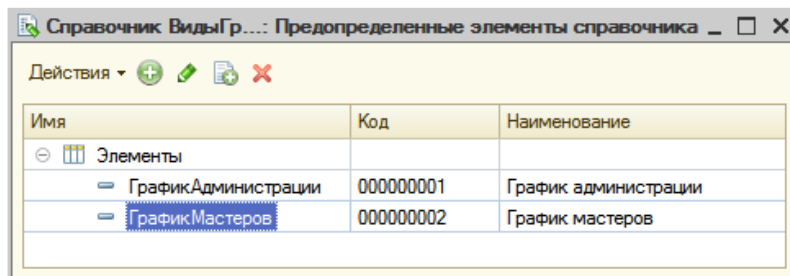


Рисунок 21 – Настройка predetermined types of calculation

5 В режиме Конфигуратор создайте новый объект «Справочник» с именем «ВидыГрафиковРаботы». На закладке Подсистемы укажите, что справочник будет отображаться в подсистеме «РасчетЗарплаты».

6 На закладке «Прочее» создайте для справочника два predeterminedных графика работы – «ГрафикАдминистрации» и «ГрафикМастеров».



Имя	Код	Наименование
Элементы		
ГрафикАдминистрации	000000001	График администрации
ГрафикМастеров	000000002	График мастеров

Рисунок 22 – Предопределенные элементы справочника «ВидыГрафиковРаботы»

7 Создадим объект конфигурации Регистр сведений с именем «ГрафикиРаботы». Этот регистр будет иметь два измерения:

- «ГрафикРаботы» типа «СправочникСсылка.ВидыГрафиковРаботы»;
- «Дата» типа «Дата».

Затем создадим единственный ресурс регистра – «Значение» типа «Число», длина 1.

На закладке «Подсистемы» укажем «РасчетЗарплаты».

8 Теперь заполним регистр сведений ГрафикиРаботы данными о рабочих днях июля графика мастеров. (Если произошла ошибка Frame.dll, настройте родительское окно в режим Соединяемое, Свободное.)

9 Запустите режим отладки и в разделе «Расчет зарплаты» перейдите в регистр сведений «Графики работы». Поочередно создайте 31 запись в регистре для июля. Для облегчения работы воспользуйтесь кнопкой «Создать новый элемент копированием текущего». В качестве измерения «ГрафикРаботы» нашего регистра выберем predeterminedный элемент «ГрафикМастеров» справочника «ВидыГрафиковРаботы». В качестве ресурса «Значение» у рабочих дней поставим 1, выходных – 0.

10 В режиме Конфигуратор добавьте новый объект конфигурации *Регистр расчета* с именем «Начисления». Зададим «Расширенное представление списка» как «Движения в регистре Начисления». В качестве плана видов расчета, используемого регистром, выберите «ОсновныеНачисления» (рис. 23).

11 Установите, что регистр будет использовать период действия, график будет задаваться в регистре сведений «ГрафикиРаботы», значение графика будет находиться в ресурсе «Значение», а дата графика – в измерении «Дата». Укажем, что регистр будет использовать базовый период и периодичностью регистра будет «Месяц».

12 На закладке Подсистемы укажем «РасчетЗарплаты».

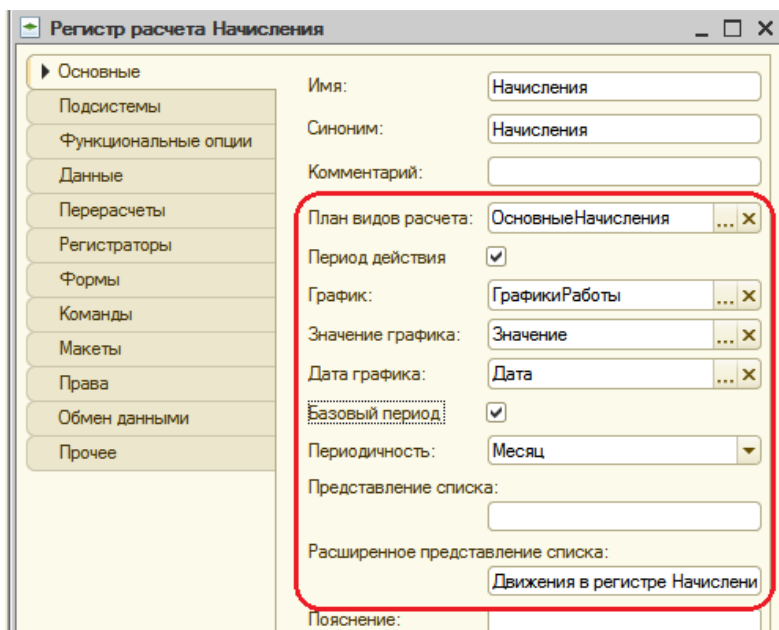


Рисунок 23 – Настройка регистра расчета

13 На закладке «Данные» создадим:

- Измерение «Сотрудник», тип «СправочникСсылка.Сотрудники», Базовое;
- Ресурс «Результат», тип «Число», длина 15, точность 2;
- Реквизит ГрафикРаботы, тип «СправочникСсылка.ВидыГрафиковРаботы», в разделе свойств «Данные» зададим свойство «Связь с графиком по измерению» ГрафикРаботы;
- Реквизит «ИсходныеДанные», тип Число, длина 15, точность 2.

Реквизит ГрафикРаботы будем использовать, чтобы связать запись регистра с используемым графиком работы, а реквизит «ИсходныеДанные» – чтобы хранить в нем данные для расчета или перерасчета.

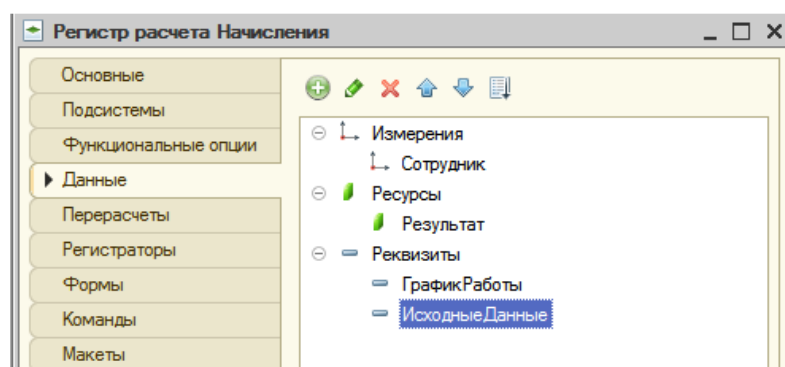


Рисунок 24 – Измерения, ресурсы и реквизиты регистра расчета

14 Перейдем на закладку «Перерасчеты». Создадим объект конфигурации «Перерасчет» с таким же именем. У него будет измерение «Сотрудник», для которого в разделе «Связь» укажем:

- Измерение регистра – «Сотрудник»;

- «Данные ведущих регистров» – выберем то же самое измерение «Сотрудник» регистра расчета «Начисления».

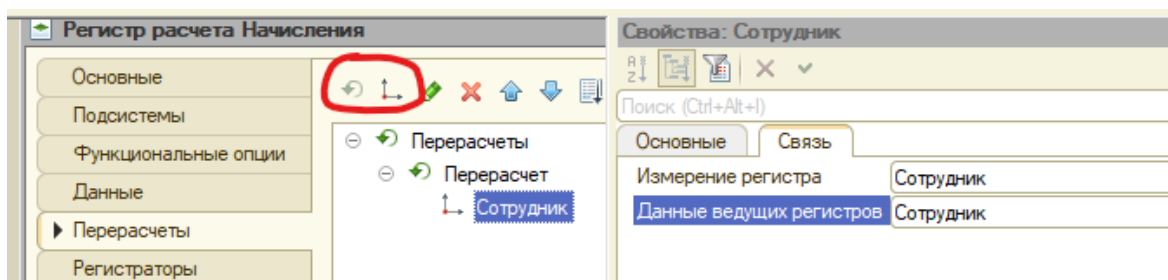


Рисунок 25 – Настройка перерасчета

15 В заключение отредактируем командный интерфейс, чтобы в подсистеме «РасчетЗарплаты» была доступна команда для просмотра записей регистра расчета. Для этого в дереве объектов конфигурации выберите подсистему «РасчетЗарплаты», вызовите контекстное меню и выберите «Открыть командный интерфейс». В группе «Панель навигации.Обычное» включите видимость команды «Начисления». На этом создание объекта «Регистр расчета» «Начисления» завершено.

Контрольные вопросы

- 1 Как использовать план видов характеристик для организации расчета.
- 2 Что такое вид расчета, база?
- 3 Какая разница между базовым периодом, периодом действия и фактическим периодом?
- 4 Что такое зависимость по базовому периоду?
- 5 Что такое вытеснение по периоду действия?
- 6 Для чего предназначен объект конфигурации «План видов расчета»?
- 7 Каковы основные свойства плана видов расчета?
- 8 Какая разница между базовыми, вытесняющими и ведущими видами расчетов?
- 9 Как создать план видов расчета?
- 10 Что такое объект конфигурации «Регистр расчета»?
- 11 Каковы отличительные особенности регистра расчета?
- 12 Что такое перерасчет?
- 13 Как создать регистр расчета?

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 7

Тема работы: Использование регистра расчета.

Цель работы: создание документов для периодических начислений.

Порядок выполнения работы

- 1 Откроем конфигуратор и добавим новый объект конфигурации «Документ» – «НачисленияСотрудникам». Зададим представление объекта как Начисление сотрудникам. На закладке Нумерация установим:

- «Тип номера» – «Число»;
- «Длина номера» – 5;
- «Контроль уникальности» – «Да»
- «Периодичность» - «Непериодический»

2 На закладке Подсистемы укажем, что документ будет отображаться в подсистеме «РасчетЗарплаты».

3 На закладке «Данные» укажем, что этот документ будет иметь табличную часть «Начисления», содержащую следующие реквизиты:

- «Сотрудник», тип «СправочникСсылка.Сотрудники»;
- «ГрафикРаботы», тип «СправочникСсылка.ВидыГрафиковРаботы»;
- «ДатаНачала», тип «Дата»;
- «ДатаОкончания», тип «Дата»;
- «ВидРасчета», тип «ПланВидовРасчетаСсылка.ОсновныеНачисления»;
- «Начислено», «Число», длина 15, точность 2.

Реквизиты «ДатаНачала» и «ДатаОкончания» понадобятся нам для того, чтобы задавать период, в котором должна действовать запись расчета.

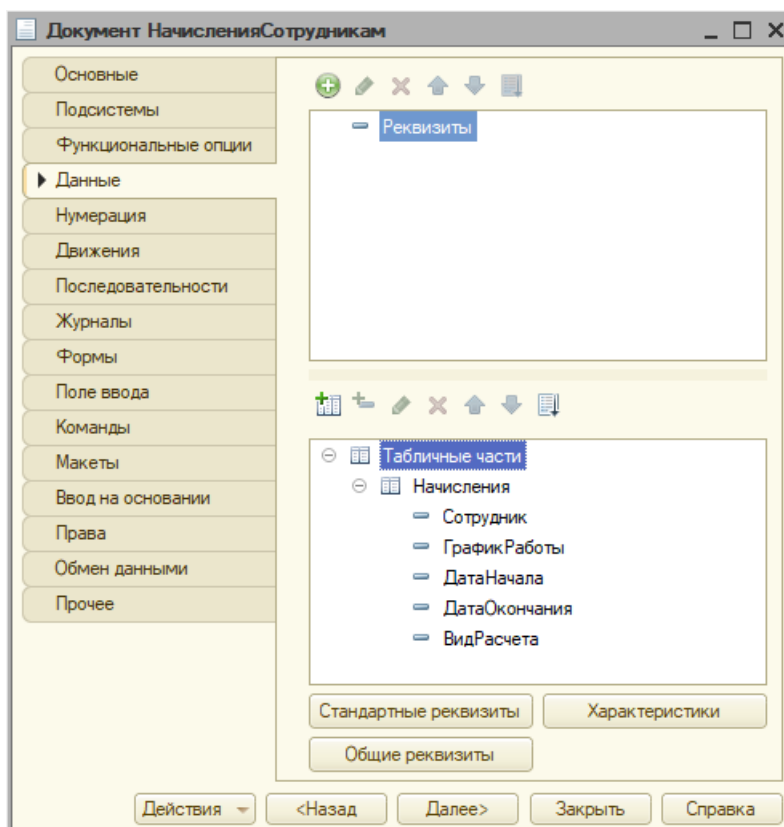


Рисунок 26 – Настройка табличной части документа «НачислениеСотрудникам»

4 На закладке «Движения» запретим оперативное проведение документа.

5 Отметим, что документ будет создавать движения по регистру расчета «Начисления», и запустим конструктор движений. В окне конструктора выберем табличную часть «Начисления» и нажмем «Заполнить выражения». Для реквизитов «ПериодДействияКонец» и «БазовыйПериодКонец» укажем выражение

КонецДня(ТекСтрокаНачисления.ДатаОкончания). Для поля «ПериодРегистрации» укажем выражение **Дата**. Реквизиту «ИсходныеДанные» поставим в соответствие реквизит табличной части «Начислено». Для ресурса Результат оставим пустое выражение, так как мы будем его потом рассчитывать. Окончательный вид окна конструктора движений представлен на рисунке 27. Нажмем «ОК» и посмотрим текст обработчика, созданный конструктором

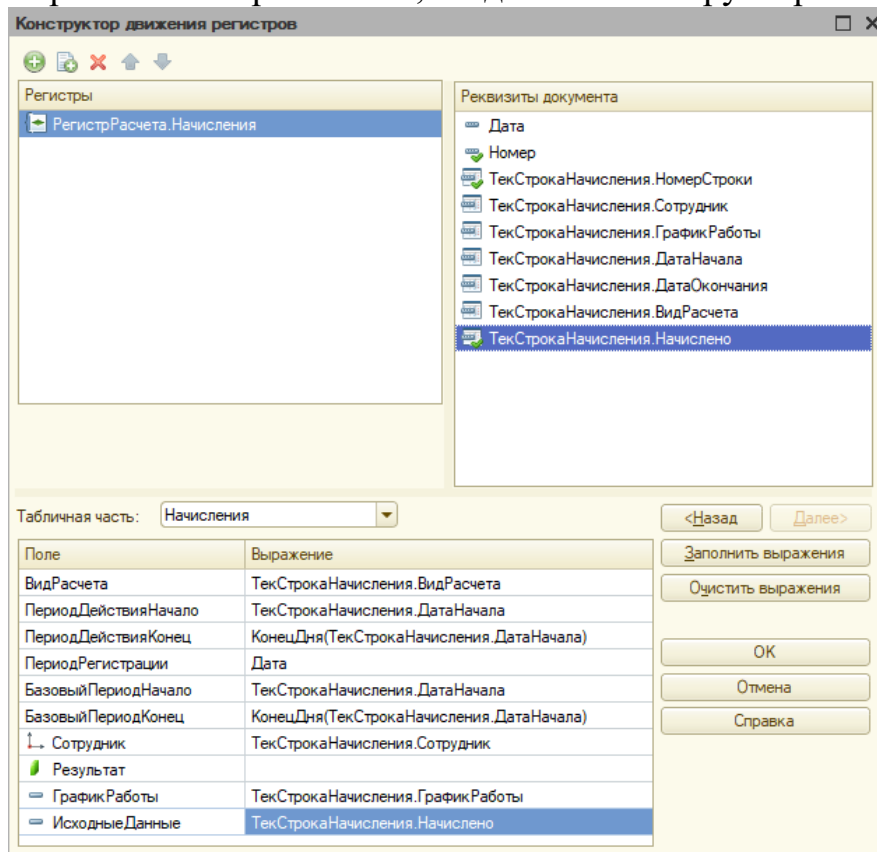


Рисунок 27 – Конструктор движений документа «НачислениеСотрудникам»

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
Движения.Начисления.Записывать = Истина;
```

```
//Регистр Начисления
```

```
Для Каждого ТекСтрокаНачисления Из Начисления Цикл
```

```
Движение = Движения.Начисления.Добавить();
```

```
Движение.Сторно = Ложь;
```

```
Движение.ВидРасчета = ТекСтрокаНачисления.ВидРасчета;
```

```
Движение.ПериодДействияНачало = ТекСтрокаНачисления.ДатаНачала;
```

```
Движение.ПериодДействияКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);
```

```
Движение.ПериодРегистрации = Дата;
```

```
Движение.БазовыйПериодНачало = ТекСтрокаНачисления.ДатаНачала;
```

```
Движение.БазовыйПериодКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);
```

```
Движение.Сотрудник = ТекСтрокаНачисления.Сотрудник;  
Движение.ГрафикРаботы = ТекСтрокаНачисления.ГрафикРаботы;  
Движение.ИсходныеДанные = ТекСтрокаНачисления.Начислено;  
КонецЦикла;  
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
КонецПроцедуры
```

6 В заключение отредактируем командный интерфейс, чтобы в разделе «Расчет зарплаты» была доступна команда создания новых документов. Для этого в дереве объектов конфигурации выделим подсистему «РасчетЗарплаты», вызовем контекстное меню и выполним команду «Открыть командный интерфейс».

7 В группе «Панель Действий.Создать» включим видимость у команды «Начисление сотрудникам: создать»

8 Запустим «1С:Предприятие» в режиме отладки и посмотрим, как работает наш документ.

9 В разделе «Расчет зарплаты» выполним команду «Начисление сотрудникам» и начислим оклад за июль всем сотрудникам

10 Проведем документ и посмотрим, какие движения он сформировал в регистре «Начисления».

Обратите внимание на то, что платформа привела период регистрации каждой записи к началу периода регистра расчета – началу месяца.

Кроме этого заметьте, что в каждой записи мы сохранили в реквизите ИсходныеДанные размер оклада сотрудника, введенный в документе, чтобы в дальнейшем рассчитать сумму оплаты по окладу.

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА 8

Тема работы: Механизм вытеснения и зависимости от базы, отчет по перерасчетам.

Цель работы: изучение дополнительных механизмов работы с регистрами расчета.

Порядок выполнения работы

1 Создадим новый объект конфигурации «Отчет». Назовем его «Перерасчет».

2 Создадим основную схему компоновки данных, добавим источник данных – запрос и откроем конструктор запроса.

3 В списке «База данных» раскроем ветвь «Перерасчеты» и из виртуальной таблицы перерасчета «Начисления.Перерасчет» выберем все поля:

- «ОбъектПерерасчета»;
- «ВидРасчета»;
- «Сотрудник».

На этом создание запроса закончено, нажмем ОК.

4 Перейдем на закладку «Настройки» и добавим группировку детальных записей.

5 На закладке «Выбранные поля» выберем для вывода в отчет поля «ОбъектПерерасчета», «ВидРасчета» и «Сотрудник». На этом создание схемы компоновки данных закончено, закроем ее.

6 В окне редактирования объекта конфигурации «Отчет Перерасчет» на закладке «Подсистемы» укажем, что отчет будет принадлежать подсистеме «РасчетЗарплаты».

7 Если сейчас мы выполним отчет в режиме «1С:Предприятие», то мы увидим, что ни один перерасчет еще не выполнялся. Поэтому создадим новый документ «Начисление сотрудникам» № 2, в котором начислим премию за июль Гусакову и Деловому

Этим документом мы зафиксируем тот факт, что сотрудникам Гусакову и Деловому нужно начислить премию по итогам работы за июль. Поскольку размер премии нам неизвестен (он будет рассчитываться по некоторому алгоритму), поле «Начислено» мы оставляем пустым. Нажмем «Провести и закрыть».

8 Теперь снова откроем документ «Начисление сотрудникам» № 1 и изменим оклад Гусакова с 10 000 на 7 000. Нажмем «Провести и закрыть».

9 Сформируем отчет «Перерасчет».

Отчет теперь содержит какие-то данные. В самом деле вид расчета «Премия» зависит у нас по базовому периоду от вида расчета «Оклад». Как только мы изменили существовавшие в регистре записи по виду расчета «Оклад», платформа сразу же сформировала набор записей перерасчета, которые должны быть рассчитаны заново, так как изменилась их база.

Платформа не отслеживает конкретные изменения, которые пользователь внес в записи документа. Она отслеживает лишь факт изменения набора записей регистра расчета в результате проведения (перепроведения) документа. Поэтому в набор записей перерасчета она включает информацию обо ВСЕХ записях регистра, значение ресурсов которых МОЖЕТ измениться в результате перепроведения документа, создавшего базовые записи регистра.

10 Перепроведем документ «Начисления сотрудникам» № 2 (которым мы начисляли премию) и сформируем отчет «Перерасчет».

11 Он снова не содержит никаких данных – система отметила тот факт, что мы «пересчитали» зависимые записи, и очистила таблицу перерасчета.

Теперь посмотрим, как работает механизм вытеснения по периоду действия.

12 Для этого нам понадобится создать документ «Начисления сотрудникам» № 3 (рис. 28).

Этим документом мы зафиксируем тот факт, что Гусаков не выходил на работу с 1 по 10 июля. Очевидно, что в этом случае потребуется пересчитать его оплату по окладу и как следствие начисленную премию. Нажмем «Провести и закрыть» и затем сформируем отчет «Перерасчет».

Провести и закрыть		Записать	Провести	Еще ▾		
Номер:	<input type="text" value="3"/>					
Дата:	<input type="text" value="16.08.2019 13:24:21"/>					
Добавить		<input type="button" value="↑"/>	<input type="button" value="↓"/>	Еще ▾		
N	Сотрудник	График работы	Дата начала	Дата окончания	Вид расчета	Начислено
1	Гусаков Иван Иванович	График мастеров	01.07.2019	10.07.2019	Невыход	

Рисунок 28 – Учет невыхода на работу

14 В перерасчет попала запись о начислении оклада Гусакову. Это явилось результатом работы механизма вытеснения по периоду действия, ведь вид расчета Невыход вытесняет у нас вид расчета Оклад. Обратите внимание, что в перерасчет попала и запись о начислении премии Гусакову. При создании предопределенных видов расчета мы указали, что результат вида расчета «Премия» будет зависеть от изменения результата вида расчета «Невыход». Эта зависимость косвенная, но поскольку явно указали такую зависимость, платформа ее отследила.

15 Перепроведем документы «Начисления сотрудникам» № 1 и № 2 и убедимся, что таблица перерасчета очистилась.

До сих пор мы с вами просто заносили в регистр расчета «Начисления» записи о том, что необходимо выполнить какой-либо вид расчета. Но каким именно образом получать эти результаты, мы не говорили. Теперь настало время описать алгоритмы формирования различных видов расчетов.

Поскольку эти алгоритмы нужно будет использовать не только в документе «Начисление сотрудникам», удобнее всего будет разместить их в отдельном общем модуле.

16 Откроем в конфигураторе текст обработчика проведения документа «НачисленияСотрудникам» и добавим в него после завершения создания движений в регистре Начисления вызов процедуры **РассчитатьНачисления()** из общего модуля ПроведениеРасчетов

Код обработчик проведения документа «НачисленияСотрудникам»:

Процедура ОбработкаПроведения(Отказ, Режим)

...

КонецЦикла;

// Записываем движения регистров

Движения.Начисления.Записать();

// Получим список всех сотрудников, содержащихся в документе

Запрос = Новый Запрос(

«ВЫБРАТЬ РАЗЛИЧНЫЕ

| НачисленияСотрудникам.Начисления.Сотрудник

| ИЗ

| Документ.НачисленияСотрудникам.Начисления


```

| КАК НачисленияСотрудникамНачисления
| ГДЕ
| НачисленияСотрудникамНачисления.Ссылка = &ТекущийДоку-
мент»);
Запрос.УстановитьПараметр(«ТекущийДокумент», Ссылка);
// Сформируем список сотрудников
ТаблЗнач = Запрос.Выполнить().Выгрузить();
МассивСотрудников = ТаблЗнач.ВыгрузитьКолонку(«Сотрудник»);
// Вызов процедуры РассчитатьНачисления из общего модуля
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисле-
ния,
ПланыВидовРасчета.ОсновныеНачисления.Оклад, МассивСотруд-
ников);
Движения.Начисления.Записать( , Истина);
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисле-
ния, ПланыВидовРасчета.
ОсновныеНачисления.Премия, МассивСотрудников);
Движения.Начисления.Записать( , Истина);
КонецПроцедуры

```

Эту процедуру мы вызываем сначала для расчета первичных записей (Оклад), а затем для расчета вторичных (Премия).

17 Теперь создадим в ветке «Общие» новый общий модуль «Проведение-Расчетов».

18 Установим флажок Вызов сервера для видимости его экспортных процедур и функций.

Добавим в него заготовку процедуры «РассчитатьНачисления».

Код процедуры «РассчитатьНачисления»:

```

Процедура РассчитатьНачисления(НаборЗаписейРегистра, Требуемый-
ВидРасчета, СписокСотрудников) Экспорт
Регистратор = НаборЗаписейРегистра.Отбор.Регистратор.Значение;
// Рассчитать первичные записи
Если ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисле-
ния.Оклад Тогда

```

```

Запрос = Новый Запрос;

```

```

Запрос.Текст =

```

```

«ВЫБРАТЬ

```

```

| НачисленияДанныеГрафика.ЗначениеПериодДействия КАК Норма,

```

```

| НачисленияДанныеГрафика.ЗначениеФактическийПериодДействия
КАК Факт,

```

```

| НачисленияДанныеГрафика.НомерСтроки КАК НомерСтроки

```

|ИЗ

| РегистрРасчета.Начисления.ДанныеГрафика(Регистратор = &Регистратор И

| ВидРасчета = &ВидРасчета И Сотрудник В (&СписокСотрудников))

| КАК НачисленияДанныеГрафика»;

Запрос.УстановитьПараметр(«Регистратор», Регистратор);

Запрос.УстановитьПараметр(«ВидРасчета», ТребуемыйВидРасчета);

Запрос.УстановитьПараметр(«СписокСотрудников», СписокСотрудников);

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

Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл

СтруктураНомер = Новый Структура(«НомерСтроки»);

СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;

ВыборкаРезультата.Сбросить();

Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда

Если ВыборкаРезультата.Норма = 0 Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = «Вид расчета: Оклад – Нет рабочих дней в заданном периоде»;

Сообщение.Сообщить();

ЗаписьРегистра.Результат = 0;

Иначе

// Рассчитать оклад по фактическому периоду и исходным данным

ЗаписьРегистра.Результат = (ЗаписьРегистра.ИсходныеДанныеВыборкаРезультата.Норма) * ВыборкаРезультата.Факт;

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = «Выполнен расчет « + ЗаписьРегистра.Регистратор + « – « + ЗаписьРегистра.ВидРасчета + « – « +

ЗаписьРегистра.Сотрудник;

Сообщение.Сообщить();

КонецЕсли;

КонецЕсли;

КонецЦикла;

// Рассчитать вторичные записи

ИначеЕсли ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Премия

Тогда

Запрос = Новый Запрос;

Запрос.Текст =

«ВЫБРАТЬ

| НачисленияБазаНачисления.РезультатБаза КАК База,

| НачисленияБазаНачисления.НомерСтроки КАК НомерСтроки

|ИЗ

```

| РегистрРасчета.Начисления.БазаНачисления(&ИзмеренияОсновного,
| &ИзмеренияБазового, , Регистратор =
| &Регистратор И ВидРасчета = &ВидРасчета И
| Сотрудник В (&СписокСотрудников))
| КАК НачисленияБазаНачисления»;
Измер = Новый Массив(1);
Измер[0] = «Сотрудник»;
Запрос.УстановитьПараметр(«ИзмеренияОсновного», Измер);
Запрос.УстановитьПараметр(«ИзмеренияБазового», Измер);
Запрос.УстановитьПараметр(«Регистратор», Регистратор);
Запрос.УстановитьПараметр(«ВидРасчета», ТребуемыйВидРасчета);
Запрос.УстановитьПараметр(«СписокСотрудников», СписокСотрудни-
ков);
ВыборкаРезультата = Запрос.Выполнить().Выбрать();
Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
СтруктураНомер = Новый Структура(«НомерСтроки»);
СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
ВыборкаРезультата.Сбросить();
Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
ЗаписьРегистра.Результат = ВыборкаРезультата.База * (10 / 100);
Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = «Выполнен расчет « + ЗаписьРегистра.Регистратор + « -
« + ЗаписьРегистра.ВидРасчета + « - « + ЗаписьРегистра.Сотруд-
ник;
Сообщение.Сообщить();
КонецЕсли;
КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Сумму начисленной премии мы рассчитываем как 10 % от рассчитанной оплаты по окладу.

19 Запустим «1С:Предприятие» в режиме отладки и проверим правильность работы процедуры расчета.

20 Отменим проведение документа «Начисления сотрудникам» № 3 («Еще» – «Отмена проведения») и перепроведем документы «Начисления сотрудникам» № 1 и № 2. В Регистре начислений мы видим, что всем сотрудникам произведены начисления по окладу (поле «Результат») за полный месяц в соответствии с исходными данными (поле «Исходные данные»). Сотрудникам Гусакову и Деловому начислена премия в размере 10 % от суммы начисления по окладу.

21 Проведем документ «Начисления сотрудникам» № 3, а затем № 1 и № 2. При этом отчет «Перерасчет» должен быть пуст.

22 В результате невыхода Гусакова на работу сумма оплаты по окладу будет уменьшена, и соответствующим образом уменьшится начисленная ему премия.

Теперь мы посмотрим, каким образом можно использовать данные, хранящиеся в регистре расчета, для получения в отчете итоговой информации о начислениях сотрудникам.

23 Создадим в конфигураторе новый объект конфигурации «Отчет».

24 Назовем его «НачисленияСотрудникам». Создадим основную схему компоновки данных отчета, добавим новый Набор данных – запрос, откроем конструктор запроса.

25 Выберем таблицу регистра расчета «Начисления». Из нее выберем следующие поля:

- Сотрудник,
- ВидРасчета
- ПериодДействияНачало
- ПериодДействияКонец
- Регистратор,
- Результат

26 На закладке «Объединения/Псевдонимы» определим следующие псевдонимы полей «ПериодДействияНачало» и «ПериодДействияКонец». На этом создание запроса закончено, нажмем «ОК».

27 Перейдем на закладку «Ресурсы» и укажем, что должна быть рассчитана сумма по полю «Результат».

28 После этого перейдем на закладку «Настройки» и создадим структуру отчета.

29 Добавим группировку по полю «Сотрудник» и в ней – подчиненную группировку детальных записей.

30 В качестве полей, выводимых в отчет, выберем поля «ВидРасчета», «Начало», «Окончание», «Регистратор» и «Результат». В результате окно настроек отчета должно иметь вид, представленный на рис. 29.

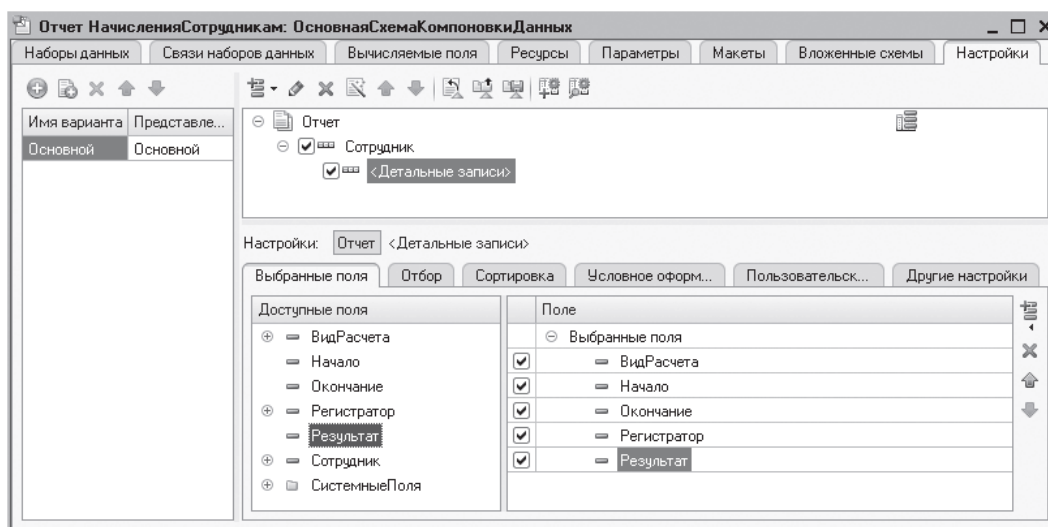


Рисунок 29 – Настройки отчета «НачисленияСотрудникам»

31 На закладке «Сортировка» укажем, что сортировка должна выполняться по возрастанию значения поля «Сотрудник» и «Регистратор».

32 На закладке «Другие настройки» зададим заголовок отчета – «Начисления сотрудникам». На этом создание схемы компоновки данных закончено, закроем ее.

33 В окне редактирования объекта конфигурации «Отчет» «НачисленияСотрудникам» на закладке подсистемы укажем, что отчет будет вызываться из подсистем «РасчетЗарплаты» и «Бухгалтерия».

34 Запустим «1С:Предприятие» в режиме отладки. В командной панели раздела Расчет зарплаты выполним команду Начисления сотрудникам и сформируем отчет

В нашем алгоритме работы с данными расчета осталось одно узкое место – контроль актуальности данных, содержащихся в регистре расчета.

До сих пор мы с вами использовали служебный отчет «Перерасчет» для того, чтобы определить, являются ли данные в регистре расчета актуальными (если отчет пуст), или же они требуют перерасчета.

Теперь мы с вами создадим специальную процедуру, которая будет определять, требуется ли перерасчет данных регистра расчета, и, если такая необходимость есть, выполнять перерасчет.

Поскольку единственным способом получения итоговой информации о начислениях сотрудникам в нашей конфигурации является отчет НачисленияСотрудникам, для вызова этой процедуры мы создадим форму этого отчета и добавим в командную панель формы кнопку «Перерассчитать», по которой будет выполняться перерасчет данных регистра.

35 Для этого в окне редактирования объекта конфигурации «Отчет» «НачисленияСотрудникам» перейдем на закладку «Формы», нажмем кнопку открытия и создадим основную форму отчета.

36 Затем в правом верхнем окне редактора форм перейдем на закладку «Команды» и на закладке «Команды формы» создадим команду формы «Перерассчитать».

37 Теперь нужно установить «Действие» для этой команды. Для этого нажмем кнопку открытия в строке «Действие». На запрос конфигууратора о типе обработчика команды ответим, что мы хотим создать обработчик, выполняющийся на клиенте.

38 В модуле формы будет создан шаблон клиентской процедуры `Перерассчитать()`, в которую мы поместим вызов процедуры `ПерерассчитатьНачисления()` из общего модуля `ПроведениеРасчетов`

Код обработчика команды «Перерассчитать»:

&НаКлиенте

Процедура Перерассчитать(Команда)

ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗначение(

«ПланВидовРасчета.ОсновныеНачисления.Оклад»));

```

ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗначение(
«ПланВидовРасчета.ОсновныеНачисления.Премия»));
КонецПроцедуры
39 Саму процедуру перерасчета поместим в общем модуле ПроведениеРасчетов (листинг 18.4).
Листинг 16.4. Процедура перерасчета начислений
Процедура ПерерассчитатьНачисления(ТребуемыйВидРасчета)
Экспорт
Запрос = Новый Запрос(
«ВЫБРАТЬ
| НачисленияПерерасчет.ОбъектПерерасчета,
| НачисленияПерерасчет.Сотрудник
| ИЗ
| РегистрРасчета.Начисления.Перерасчет КАК НачисленияПерерасчет
| ГДЕ
| НачисленияПерерасчет.ВидРасчета = &ТребуемыйВидРасчета
| ИТОГИ ПО
| НачисленияПерерасчет.ОбъектПерерасчета»);
Запрос.УстановитьПараметр(«ТребуемыйВидРасчета», ТребуемыйВидРасчета);
СписокСотрудников = Новый СписокЗначений;
// Перебрать группировку по регистратору.
ВыборкаПоРегистратору = Запрос.Выполнить().Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
Пока ВыборкаПоРегистратору.Следующий() Цикл
Регистратор = ВыборкаПоРегистратору.ОбъектПерерасчета;
// Перебрать группировку по сотрудникам для выбранного регистратора
// и создать список сотрудников.
ВыборкаПоСотрудникам = ВыборкаПоРегистратору.Выбрать();
СписокСотрудников.Очистить();
Пока ВыборкаПоСотрудникам.Следующий() Цикл
СписокСотрудников.Добавить(ВыборкаПоСотрудникам.Сотрудник);
КонецЦикла;
// Получить набор записей регистра расчета для выбранного регистратора.
НаборЗаписей = РегистрыРасчета.Начисления.СоздатьНаборЗаписей();
НаборЗаписей.Отбор.Регистратор.Значение = Регистратор;
НаборЗаписей.Прочитать();

```

```

РассчитатьНачисления(НаборЗаписей, ТребуемыйВидРасчета, Сп-
сокСотрудников);
НаборЗаписей.Записать( , Истина);
// Очистить перерассчитанные записи в перерасчете.
НаборЗаписейПерерасчета =
РегистрыРасчета.Начисления.Перерасчеты.Перерасчет.Со-
здатьНаборЗаписей();
НаборЗаписейПерерасчета.Отбор.ОбъектПерерасчета.Значение =
Регистратор;
НаборЗаписейПерерасчета.Записать();
КонецЦикла;
КонецПроцедуры

```

40 Вернемся в форму отчета «НачислениеСотрудникам».

41 Перетащим мышью команду «Перерассчитать» в группу элементов формы «ОсновнаяКоманднаяПанель». При этом в форме появится кнопка «Перерассчитать», а связь кнопки с командой будет установлена автоматически.

42 Запустим «1С:Предприятие» и проверим, как выполняется перерасчет записей регистра расчета.

43 Отменим проведение всех документов «Начисления сотрудникам» и проведем документ «Начисления сотрудникам» № 1 и затем № 2.

44 Сформируем отчет «Начисления сотрудникам».

45 Теперь откроем документ «Начисления сотрудникам» № 1, изменим оклад Гусакова на 10 000 и проведем документ.

46 В отчете «Начисления сотрудникам» нажмем кнопку «Перерассчитать». Будет выполнен перерасчет начисления премии Гусакову и Деловому.

47 Чтобы увидеть в отчете актуальные данные, нажмем кнопку «Сформировать». Результат работы отчета будет содержать новые значения премии Гусакова.

48 И, наконец, проведем документ «Начисления сотрудникам» № 3 и нажмем «Перерассчитать» в отчете «Начисления сотрудникам». Снова будет произведен перерасчет оклада и премии Гусакова

49 Затем нажмем кнопку «Сформировать». Данные отчета будут содержать актуальные значения начисления оклада и премии

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 1С:Предприятие [Электронный ресурс]. – М., 1997–2019. – Режим доступа : <http://v8.1c.ru> (дата обращения: 01.06.2019).
- 2 8.2. Язык запросов 1С:Предприятие 8.2. Руководство [Электронный ресурс]. – М., 1997–2019. – Режим доступа : http://bibliopole.biz/buhgalterskiy-uchet_814/yazyik-zaprosov-23817.html (дата обращения: 28.01.2017)
- 3 1С_Предприятие 8.3. Руководство разработчика [Электронный ресурс]. – М., 2012. – Режим доступа : <http://mexalib.com/download/29105#1> (дата обращения: 01.06.2019).
- 4 **Мельников, П.П.** Разработка учетных и аналитических приложений в среде 1С:Предприятие [Электронный ресурс] / П.П. Мельников. – М., 2011. – Режим доступа : <http://docplayer.ru/33164000-Razrabotka-uchetnyh-i-analiticheskikh.html> (дата обращения: 01.06.2019).