

Лабораторная работа 8. Периодические регистры сведений

Объект конфигурации **Регистр сведений** предназначен для описания структуры хранения данных в разрезе нескольких измерений. На основе объекта конфигурации **Регистр сведений** платформа создает в базе данных таблицу, в которой может храниться произвольная информация, **привязанная** к набору измерений.

Важной особенностью регистра сведений является его способность хранить данные с привязкой ко времени. Благодаря этому регистр сведений может хранить не только актуальные значения данных, но и историю их изменения во времени. Регистр сведений, использующий привязку ко времени, называют **периодическим регистром сведений**.

Периодический регистр сведений всегда содержит служебное поле **Период**, добавляемое системой автоматически. Оно имеет тип **Дата** и служит для указания факта принадлежности записи к какому-либо периоду. При записи данных в регистр платформа всегда приводит значение этого поля к началу того периода, в который он попадает.

Система контролирует уникальность записей для регистра сведений.

Ключом записи, однозначно идентифицирующим запись, является совокупность значений измерений регистра и периода, если регистр сведений периодический. Регистр сведений не может содержать несколько записей с одинаковыми ключами.

Наряду с возможностью использования в режиме подчинения регистратору, когда записи **привязаны** к документу-регистратору, регистр сведений может применяться и в независимом режиме, в котором пользователю предоставляется полная свобода интерактивной работы с данными регистра. Регистр сведений, не использующий подчинение регистратору, называют **независимым регистром сведений**.

Добавление периодического регистра сведений

Периодический регистр сведений будет хранить развернутые во времени розничные цены материалов и стоимости услуг, оказываемых предприятием.

В режиме Конфигуратор

Добавим стандартным способом новый объект конфигурации **Регистр сведений** в ветви **Регистры сведений** дерева объектов конфигурации. В открывшемся окне редактирования объекта на закладке **Основные** введем имя регистра – **Цены**. Установим свойство **Периодичность** этого регистра – **В пределах месяца**. Зададим свойства **Представление записи** как **Цена**, а **Представление списка** как **Цены на номенклатуру**.

Свойство **Режим записи** по умолчанию имеет значение – **Независимый**, то есть мы создаем независимый регистр сведений и сможем в дальнейшем вводить в него данные без использования регистратора – **вручную**.

На закладке **Подсистемы** отметим, что созданный регистр должен быть доступен в разделах **Учет материалов**, **Оказание услуг** и **Бухгалтерия**.

Перейдем на закладку **Данные** и создадим измерение **Номенклатура** с типом **СправочникСсылка.Номенклатура**. Укажем, что это измерение будет ведущим (рис. 8.1).

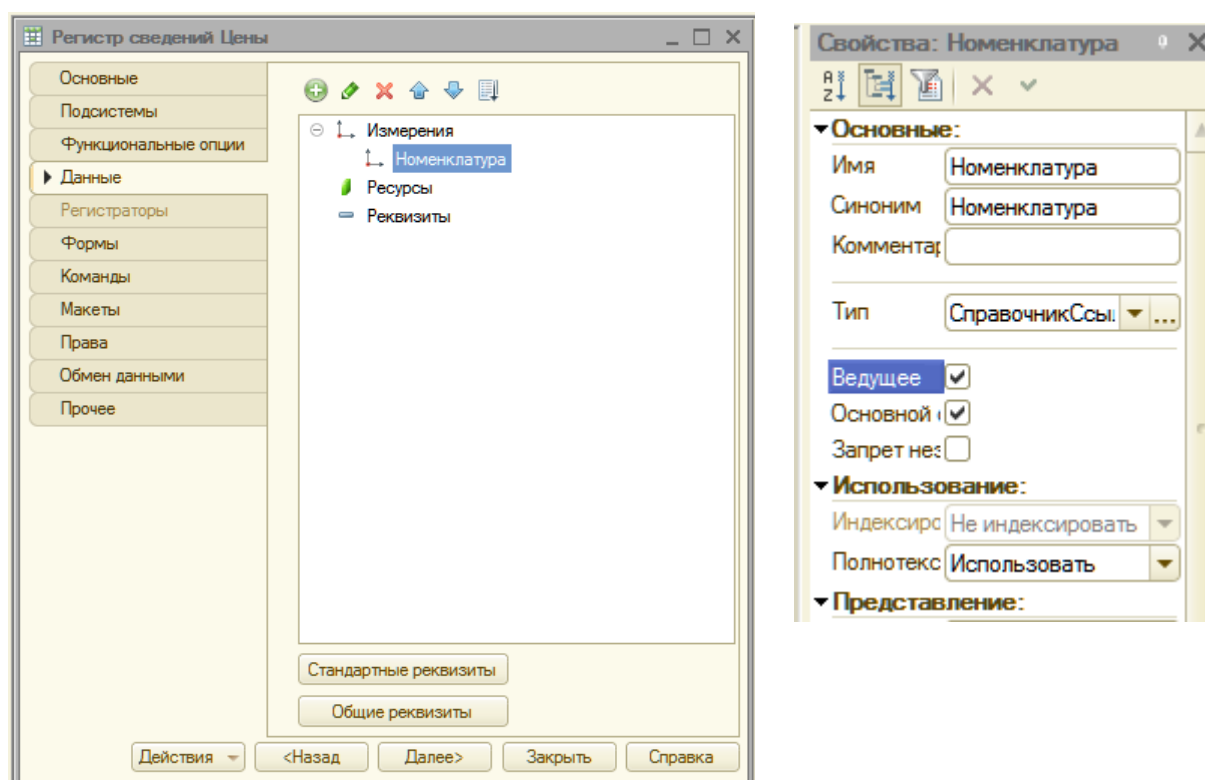


Рис. 8.1. Создание ведущего измерения регистра сведений

Свойство **Ведущее** используется тогда, когда измерение имеет тип ссылки на объект базы данных. Установка свойства **Ведущее** свидетельствует о том, что запись регистра сведений представляет интерес, пока существует тот объект,

ссылка на который выбрана в качестве значения этого измерения. При удалении объекта, все записи регистра сведений по этому объекту тоже будут автоматически удалены.

Так как это измерение регистра сделано ведущим, то в форме элемента справочника **Номенклатура** в панели навигации в группе **Перейти** появится ссылка. По ней возможен переход к записям этого регистра, содержащим в измерении **Номенклатура** ссылку на этот элемент справочника.

Затем создадим ресурс **Цена**, тип **Число**, длина 15, точность 2, неотрицательное.

В режиме 1С:Предприятие

Запустим 1С:Предприятие в режиме отладки и посмотрим, как работает созданный периодический регистр сведений **Цены**.

В открывшемся окне 1С:Предприятия видно, что в панели навигации разделов **Бухгалтерия**, **Оказание услуг** и **Учет материалов** появилась команда для открытия списка регистра **Цены на номенклатуру** (рис. 8.2).

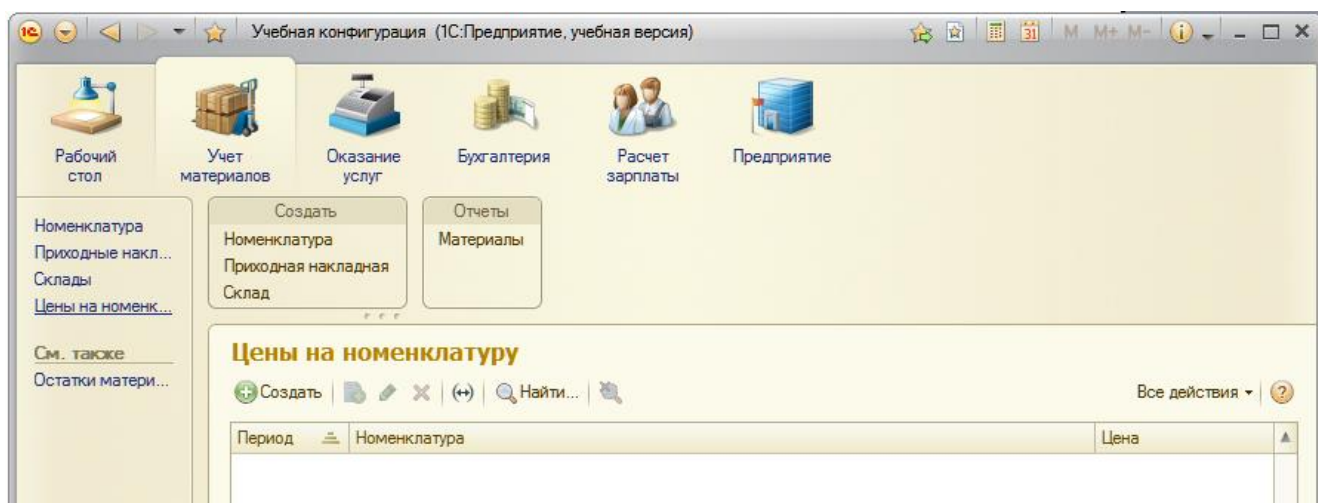


Рис. 8.2. Команда открытия регистра сведений **Цены**

Команда открытия регистра сведений по умолчанию доступна в интерфейсе разделов, в которых отображается регистр. В отличие от регистров накопления в регистрах сведений предполагается изменение данных пользователем.

Создание записей в регистре сведений

Чтобы добавить новую запись в регистр сведений, нажмем кнопку **Создать**. Зададим стоимость услуг и материалов (рис. 8.3).

Период зададим задним числом, так как он должен быть меньше или равен дате создания документа об оказании услуг **10.07.2014**.

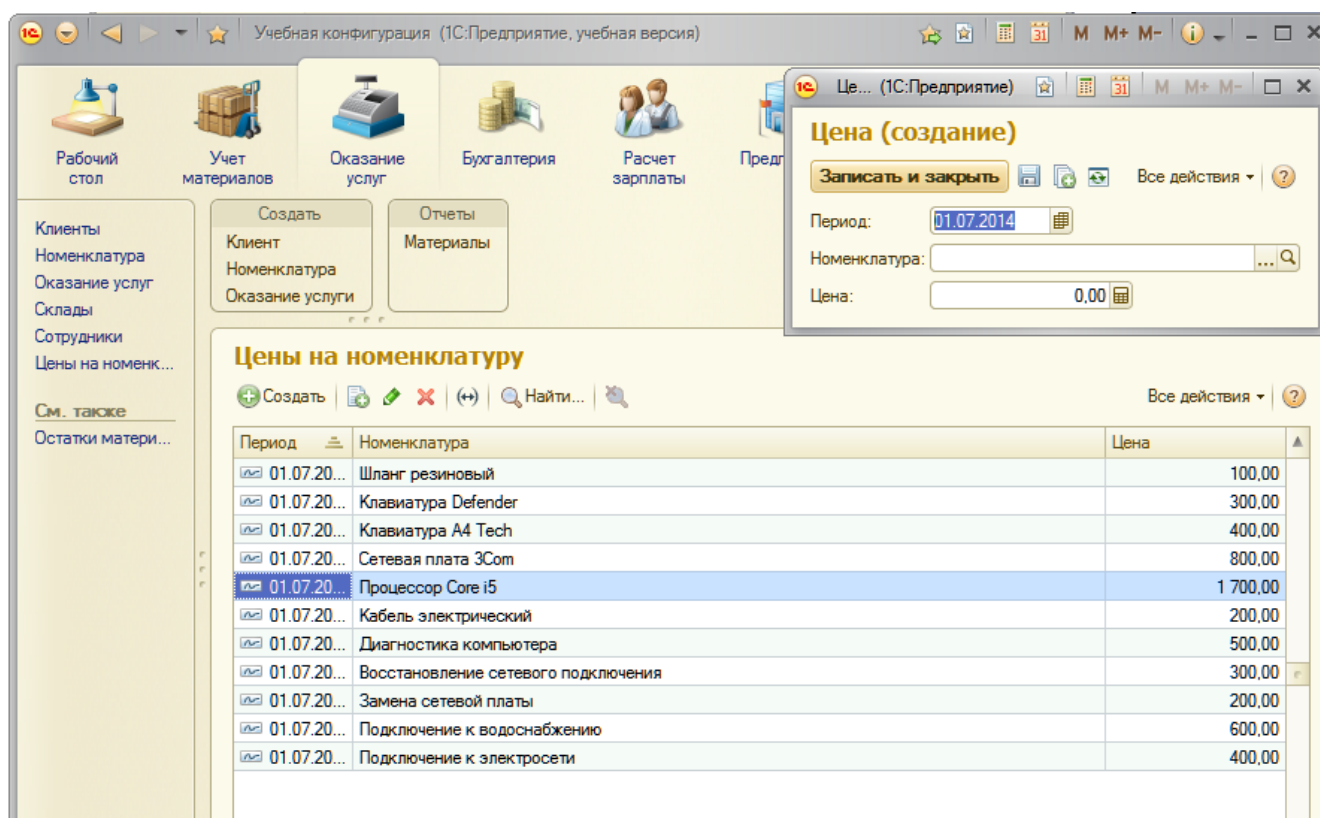


Рис. 8.3. Цены на материалы и услуги в регистре сведений Цена

Поскольку цены хранятся с привязкой к дате, мы можем заранее установить новые цены и быть уверены в том, что новые цены вступят в силу не раньше указанного для них срока.

Автоматическая подстановка цены в документ при выборе номенклатуры

Цены номенклатуры теперь хранятся в отдельном регистре сведений. Когда мы создаем или изменяем документ **ОказаниеУслуги** и добавляем в табличную часть какую-либо номенклатуру, желательно, чтобы одновременно с этим в документ подставлялась сразу и актуальная цена этой номенклатуры, полученная из регистра сведений и соответствующая дате документа.

Для этого нам нужно сделать следующее. **Сначала** написать функцию, которая будет возвращать актуальную цену номенклатуры, и затем вызвать эту функцию в тот момент, когда в документ добавляется номенклатура. В документ подставится цена номенклатуры, которую вернет эта функция.

И второе. Поскольку такая процедура понадобится нам, вероятно, не только в этом документе, но и в других документах, которые содержат в табличной части номенклатуру, мы поместим эту функцию в общий модуль.

В режиме Конфигуратор

Функция, возвращающая цену номенклатуры

Создадим функцию **РозничнаяЦена()**, которая будет возвращать актуальную розничную цену номенклатуры, и поместим ее в общий модуль конфигурации.

Откроем конфигуратор, в ветке **Общие|Общие модули** добавим новый объект конфигурации **Модуль** и назовем его **РаботаСУслугами**.

У модуля по умолчанию установлен флажок **Сервер**. Это означает, что экземпляры этого модуля будут скомпилированы только на стороне сервера. Так как сервера у нас нет, то вместо флажка **Сервер** установим флажок **Клиент** (управляемое приложение) для того, чтобы экспортные процедуры и функции этого модуля выполнялись на клиенте (рис. 8.4).

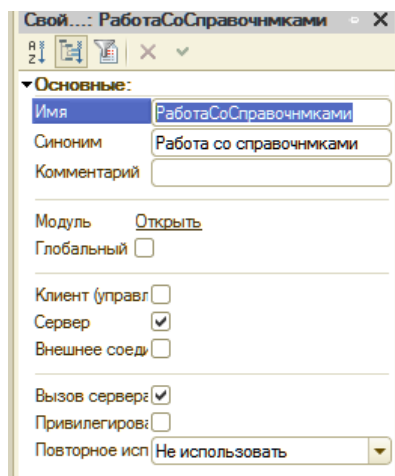


Рис. 8.4. Свойства общего модуля

Помести в модуль следующий текст.

Функция РозничнаяЦена(АктуальнаяДата, ЭлементНоменклатуры) Экспорт

// Создать вспомогательный объект Отбор

Отбор = Новый Структура ("Номенклатура", ЭлементНоменклатуры);

// Получить актуальные значения ресурсов регистра,

ЗначенияРесурсов=

РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,

Отбор);

Возврат ЗначенияРесурсов.Цена;

КонецФункции

Для получения розничной цены мы будем передавать в функцию два параметра:

✓ **АктуальнаяДата** – параметр типа **Дата** определяет точку на оси времени, для которой необходимо найти значение розничной цены;

✓ **ЭлементНоменклатуры** – ссылка на элемент справочника **Номенклатура**, для которого нужно получить розничную цену.

В теле функции сначала создается вспомогательный объект **Отбор**.

Это структура, содержащая отбор по измерениям регистра. С его помощью определяем, что необходимо найти записи регистра, в которых измерение регистра **Номенклатура** равно переданной в функцию ссылке на элемент справочника. Имя ключа структуры "**Номенклатура**" должно совпадать с именем измерения регистра, заданного в конфигураторе, а значение элемента структуры **ЭлементНоменклатуры** задает отбираемое по данному измерению значение.

Во второй строке мы обращаемся к менеджеру регистра сведений **РегистрыСведений.Цены** и выполняем метод **ПолучитьПоследнее()**, который возвращает значения ресурсов самой поздней записи регистра, соответствующей передаваемой в функцию дате – **АктуальнаяДата** и значениям измерений регистра – **Отбор**.

Метод **ПолучитьПоследнее** возвращает структуру, содержащую значения ресурсов. Структура сохраняется в переменной **ЗначенияРесурсов**. В общем случае, у регистра может быть несколько ресурсов. В нашем регистре ресурс один, поэтому будет возвращена структура, содержащая единственный элемент.

В следующей строке мы получаем искомую розничную цену, указав имя нужного ресурса регистра через точку – **ЗначенияРесурсов.Цена**. Искомая розничная цена возвращается при выполнении функции.

Теперь эту функцию следует вызвать в нужный момент работы документа.

Вызов функции при выборе номенклатуры и заполнение цены в документе

При редактировании документа **ОказаниеУслуги** необходимо обеспечить автоматическое заполнение поля **Цена** после того, как **пользователь выберет услугу**. Причем цена услуги должна определяться исходя из даты создаваемого

документа.

Откроем в конфигураторе форму документа **ОказаниеУслуги**. Откроем палитру свойств элемента формы **ПереченьНоменклатурыНоменклатура**, дважды щелкнув на элементе, или в контекстном меню откроем его **Свойства**. В конце списка найдем событие **ПриИзменении**, которое возникает после изменения значения поля.

Нажмем кнопку открытия со значком лупы в поле ввода.

Система создаст шаблон процедуры обработчика этого события в модуле формы и откроет закладку **Модуль редактора формы**.

Внесем в модуль следующий текст.

//Получить текущую строку табличной части

СтрокаТабличнойЧасти=

Элементы.ПереченьНоменклатуры.ТекущиеДанные;

// Установить цену.

СтрокаТабличнойЧасти.Цена =

РаботаСУслугами.РозничнаяЦена(Объект.Дата,

СтрокаТабличнойЧасти.Номенклатура);

// Пересчитать сумму строки.

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

Первая строка обработчика нам уже знакома по процедурам **ПереченьНоменклатурыКоличествоПриИзменении** и **ПереченьНоменклатурыЦенаПриИзменении**. Сначала мы получаем текущую строку табличной части документа, так как она нам понадобится в дальнейшем, и сохраняем ее в переменной **СтрокаТабличнойЧасти**.

Затем мы вызываем функцию **РозничнаяЦена()** из общего модуля **РаботаСоСправочниками**. Первым параметром мы передаем в эту функцию дату документа, на которую необходимо получить цену. Дату документа мы получаем из основного реквизита формы – **Объект.Дата**. Вторым параметром мы передаем ссылку на элемент справочника **Номенклатура**, который содержится в текущей строке табличной части документа – **СтрокаТабличнойЧасти.Номенклатура**.

Функция возвращает последнее значение цены, и это значение мы присваиваем полю **Цена** в текущей строке табличной части документа – **СтрокаТабличнойЧасти.Цена**.

Затем мы вызываем процедуру **РассчитатьСумму** из общего модуля **РаботаСДокументами**. Эту процедуру создали ранее для того, чтобы при изменении цены или количества в документе пересчитывать сумму в строке документа автоматически.

Процедура **ПереченьНоменклатурыНоменклатураПриИзменении()** начинает работать в модуле формы на стороне клиента, так как это обработчик интерактивного события формы. Создавая заготовку этой процедуры, платформа автоматически разместила перед описанием процедуры директиву компиляции **&НаКлиенте**.

Затем вызывается функция **РозничнаяЦена()**. Исполнение этой функции будет передано в общий модуль **РаботаСУслугами**. После завершения функции программный код продолжит исполняться в модуле формы.

Если посмотреть в синтакс-помощнике описание метода **ПолучитьПоследнее()** регистра сведений, то можно заметить, что этот метод доступен только на сервере, в **толстом клиенте** и во внешнем соединении.

Толстый клиент и внешнее соединение – это клиентские приложения прежней версии платформы, которые существуют для совместимости с прежними прикладными решениями.

Мы разрабатываем совершенно новое прикладное решение, которое может работать в тонком клиенте или в веб-клиенте. Поэтому для получения какой-либо информации из базы данных нужно передать исполнение кода на сервер, там получить нужные данные и вернуть эти данные клиенту. Это мы и сделали.

В режиме 1С:Предприятие

Проверим, как теперь работает документ **ОказаниеУслуги**.

Запустим 1С:Предприятие в режиме отладки и откроем регистр сведений **Цены**.

Для процессора Intel Core i5 добавим другим числом новую цену (**рис. 8.5**).

Теперь откроем документ **Оказание услуги № 1**. Этим документом мы как раз **израсходовали** 3 таких процессора.

Оставим дату документа без изменения и повторим выбор процессора в

Цены на номенклатуру		
<div> Создать Найти... </div> <div>Все действия ▾</div>		
Период	Номенклатура	Цена
01.07.20...	Шланг резиновый	100,00
01.07.20...	Клавиатура Defender	300,00
01.07.20...	Клавиатура A4 Tech	400,00
01.07.20...	Сетевая плата 3Com	800,00
01.07.20...	Процессор Core i5	1 700,00
01.07.20...	Кабель электрический	200,00
01.07.20...	Диагностика компьютера	500,00
01.07.20...	Восстановление сетевого подключения	300,00
01.07.20...	Замена сетевой платы	200,00
01.07.20...	Подключение к водоснабжению	600,00
01.07.20...	Подключение к электросети	400,00
10.07.20...	Сетевая плата 3Com	850,00
13.07.20...	Процессор Core i5	1 600,00

Рис. 8.5. Регистр сведений Цены

колодке **Номенклатура** табличной части документа с новым значением цены процессора (рис. 8.6).

Оказание услуги 000000001 от 22.06.2014 17:10:05 - Учебная конфигурация * (1С:Предприятие)

Оказание услуги 000000001 от 22.06.2014 17:10:05 *

Провести и закрыть Провести Печать Все действия ▾ ?

Номер: 000000001

Дата: 22.06.2014 17:10:05

Склад: Основной

Клиент: Бендер Остап Ибрагимович

Мастер: Куприн Александр Иванович

Добавить

N	Номенклатура	Количество	Цена	Сумма
1	Процессор Core i5	3,000		

Рис. 8.6. Заполнение документа Оказание услуги

Теперь установим новую дату документа и снова повторим выбор транзистора. Будет установлено новое значение цены на эту дату (рис. 8.7). В документе появляется актуальная на момент создания документа цена услуги.

Оказание услуги 000000001 от 14.07.2014 21:49:55 - Учебная конфигурация * (1С:Предприятие)

Оказание услуги 000000001 от 14.07.2014 21:49:55 *

Провести и закрыть Провести Печать Все действия ▾ ?

Номер: 000000001

Дата: 14.07.2014 21:49:55

Склад: Основной

Клиент: Бендер Остап Ибрагимович

Мастер: Куприн Александр Иванович

Добавить

N	Номенклатура	Количество	Цена	Сумма
1	Процессор Core i5	3,000	1 600,00	4 800,00
Всего:				4 800,00

Рис. 8.7. Документ Оказание услуги с новой ценой услуги

Контрольные вопросы

1. Для чего предназначен объект конфигурации Регистр сведений?
2. Какими особенностями обладает объект конфигурации Регистр сведений?
3. В чем главные отличия регистра сведений от регистра накопления?
4. Какие поля определяют ключ уникальности регистра накопления?
5. Что такое периодический регистр сведений и что такое независимый регистр сведений?
6. Как создать периодический регистр сведений?
7. Что такое ведущее измерение регистра?
8. Как получить значения ресурсов наиболее поздних записей регистра средствами встроенного языка?