

Класс Регистры

Детали хранения информации в регистрах

Регистр сведений, накопления, бухгалтерии, расчета относятся к необъектным сущностям. Примеры для регистра накопления и регистра сведений относятся ко всем классам регистров. Для регистра сведений есть возможность использования без регистратора, поэтому модель его использования шире.

Модель хранения данных. В базе данных хранятся записи. Запись не является объектом базы данных, нет внутреннего идентификатора. Запись полностью описывается значениями своих полей. Если удалив некоторую запись и записав новую с точно такими же значениями всех полей, мы получим то же состояние базы данных с точки зрения логики прикладного решения. Это принципиально отличает запись от объекта в объектных сущностях, так как объект базы данных имеет внутренний идентификатор и один и тот же объект нельзя создать дважды.

Например, если у нас есть регистр накопления, хранящий движения складских запасов товаров в разрезе товаров и складов, то каждая запись полностью определяется товаром, складом и количеством. Действительно, с точки зрения логики прикладного решения такая запись описывает движение некоторого товара по некоторому складу в некотором количестве. Можно записать вторую такую же запись, и они ничем не будут отличаться, если не отличаются значения полей.

Уникальность записей. Нельзя записывать ссылки на записи в поля базы данных, но есть уникальный ключ. Для необъектных сущностей с регистратором уникальный ключ включает регистратор и номер строки. Номер строки используется для обеспечения уникальности записей и для упорядочивания записей в пределах регистратора.

Для регистров сведений не подчиненных регистратору уникальным ключ это совокупность измерений и периода, если регистр периодический. В случае подчинения регистратору уникальность поддерживается по сочетанию полей регистратор и номер строки и по измерениям и периоду. Первое определяется его подчинением регистратору, а второе его основной прикладной логикой – хранением значений ресурсов по комбинации значений измерений и периоду. Если для регистра выбрана периодичность по позиции регистратора, то в уникальный ключ по измерениям входит и регистратор, как дополнительная детализация периода в пределах секунды.

Подчинение регистратору. Записи регистров подчиненные регистратору называются движениями. Подчинение записей регистратору определяет время жизни записей. Регистратор определяет наличие этих записей. Записи регистров создаются документом в процессе проведения. При удалении регистратора подчиненные ему записи удаляются.

Записи регистров подчинены регистратору, но не являются его частью. При считывании документа подчиненные ему записи регистров не считываются, а при записи автоматически не записываются. Обеспечивается только удаление записей при удалении регистратора. Также существует возможность автоматического удаления записей при проведении и отмене проведения. Эта возможность настраивается в свойствах документа в конфигурации и является сервисной.

Наличие записей регистров не связано с состоянием "проведен" и с пометкой удаления. Допускается наличие записей и у непроведенного документа и у помеченного на удаление. Например, это используется для создания документов предназначенных для непосредственного ввода движений регистров – ручного ввода. В этом случае пользователь непосредственно вводит записи регистров и понятие проведения бессмысленно. При непосредственном изменении полей Проведен и ПометкаУдаления (без использования методов УстановитьПометкуУдаления() и записи с режимом ОтменаПроведения) удаление движений не производится.

Поэтому организация редактирования движений документа в форме сложнее, чем организация редактирования табличной части. Наиболее простой моделью использования регистров является создание движений при проведении. В этом случае достаточно реализовать запись движений в обработчике проведения, а вся остальная логика будет поддерживаться системой автоматически.

Типы данных. Тип РегистрыНакопленияМенеджер предназначен для доступа к менеджерам конкретных регистров. К нему можно обратиться с помощью свойства глобального контекста РегистрыНакопления.

Тип РегистрНакопленияМенеджер предоставляют доступ к общим действиям, относящимся к конкретному регистру накопления. С помощью его методов выполняются действия, относящиеся к регистру, а не к его конкретным записям. Например, методы менеджера позволяют создать объект, предназначенный для манипулирования записями (НаборЗаписей), получить выборку регистра и т.д.

Менеджер регистра имеет набор методов для выполнения действий связанных с основным назначением регистра. Например, для регистра накопления это получения остатков и оборотов, а у регистра сведений получение среза первых и среза последних записей. Также данные могут быть получены виртуальными таблицами запросов, в них обеспечивается большая гибкость.

Тип РегистрНакопленияНаборЗаписей предназначен для чтения, модификации и удаления записей регистра. Набор записей оперирует множеством записей отбираемых по некоторому условию. Для описания условия есть свойство Отбор. Для регистров подчиненных регистратору отбор устанавливается по регистратору.

Набор записей это коллекция, которую можно прочитать и записать. Набор записей имеет только один метод для изменения данных – Записать(). Нет понятия удаления. Набор записей может быть только записан. При этом выполняется замещение всех существующих записей удовлетворяющих текущему отбору на записи, хранящиеся в наборе. Для удаления множества записей следует установить отбор и, не добавляя записей в набор, выполнить запись набора.

Можно записать набор без замещения, параметр метода Записать(). Запись без замещения выполняет добавление записей. Будет обеспечиваться корректировка номеров строк, для обеспечения последовательной нумерации записей в пределах регистратора. Обычно используется при необходимости записи большого объема движений регистров.

Набор записей необязательно считывать, чтобы выполнить запись. Не существует понятия блокировки, поэтому при записи набора могут быть изменены данные, которые были изменены другой сессией или другим набором в этой сессии уже после считывания набора записей.

Для регистров сведений не подчиненных регистратору может использоваться отбор с различными комбинациями измерений и периода. Допускается чтение и запись набора без установки отбора. В этом случае будет выполняться чтение и соответственно запись всех записей регистра. При ошибках в написании модулей (например, пропущенном вызове чтения набора или пропущенной установке отбора) запись набора может привести к полной очистке регистра, так как запись при неустановленном отборе фактически замещает весь регистр на содержимое набора записей.

Запись в наборе записей регистра накопления имеет тип РегистрНакопленияЗапись. Данный тип используется только для записи в наборе записей. Это значение не используется отдельно от набора записей.

Для регистра сведений не подчиненного регистратору кроме набора записей существует тип РегистрСведенийМенеджерЗаписи. Этот тип предназначен для чтения и записи одной записи регистра. Используется для организации редактирования одной записи в форме. Позволяет прочитать запись с определенными значениями ключевых полей, изменить поля, в том числе и ключевые, и записать. Является вспомогательным. Для работы с регистром он использует два набора записей (с отборами соответствующими соответственно ключевым полям считанной и записываемой записи). Соответственно реализация обработчиков в модуле набора записей позволят полностью контролировать изменения данных регистра, в том числе и тогда, когда они выполняются с помощью менеджера записи.

Тип РегистрНакопленияКлючЗаписи предназначен для уникальной идентификации записи регистра. Например, он используется для идентификации строки табличного поля отражающего список регистра. Это позволяет активизировать необходимую строку. Свойства ключа определяются полями образующими уникальный ключ регистра. Для регистров сведений это измерения, период и регистратор, если используется периодичность по позиции регистратора. Для других регистров это регистратор и номер строки. Ключ записи регистра может быть создан с помощью менеджера.

Тип РегистрНакопленияВыборка предназначен для динамического обхода записей регистра. Механизм выборок для необъектных таблиц не имеет существенных отличий. Следует учитывать, что выборка всегда считывает данные записей целиком (считываются все поля).

Тип РегистрНакопленияСписок предназначен для динамического просмотра записей регистра в элементе управления ТабличноеПоле. У него может быть настроен состав

считываемых полей (колонок), отбор и сортировка. Список осуществляет считывание данных порциями в процессе навигации пользователем в табличном поле.

Кроме указанных типов значений регистр определяет несколько расширений элементов управления и форм, предназначенных для интерактивного ввода и просмотра, данных регистра. Расширения не являются типами данных, а добавляют специфические свойства, методы и события к соответствующим объектам. Кроме того, расширения определяют некоторое специфическое поведение форм и элементов управления при конфигурировании и работе пользователя с системой.

Для регистров сведений кроме расширений для показа в форме списка и набора записей поддерживается расширение для редактирования в форме одной записи. В этом случае в качестве основного реквизита формы выступает менеджер записи регистра сведений.

Оборотный регистр накопления

Создается только одна виртуальная таблица "Обороты". Остатки не регистрируются. Тип настраивается во вкладке Основное.

Регистр накопления

Движения. Накапливает числовой итог в разрезе измерений по ресурсам. Остатки - только итоговое значение, обороты - состояние регистра на конкретную дату.

Необходимо указать регистраторы и затем в настройках каждого документа создать обработчик проведения. Можно через конструктор движения. Пример обработки проведения в модуле объекта документа

```
Процедура ОбработкаПроведения(Отказ, Режим)
```

```
□/{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
□/ Данный фрагмент построен конструктором.
```

```
□/ При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
```

```
□/ регистр ОстаткиКанцелярии Приход
```

```
□Движения.ОстаткиКанцелярии.Записывать = Истина;
```

```
□Для Каждого ТекСтрокаТаблицаПоступленияКанцелярии Из ТаблицаПоступленияКанцелярии Цикл
```

```
□□Движение = Движения.ОстаткиКанцелярии.Добавить();
```

```
□□Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
```

```
□□Движение.Период = Дата;
```

```
□□Движение.Предмет = ТекСтрокаТаблицаПоступленияКанцелярии.НазваниеПредмета;
```

```
□□Движение.Количество = ТекСтрокаТаблицаПоступленияКанцелярии.Количество;
```

```
□КонецЦикла;
```

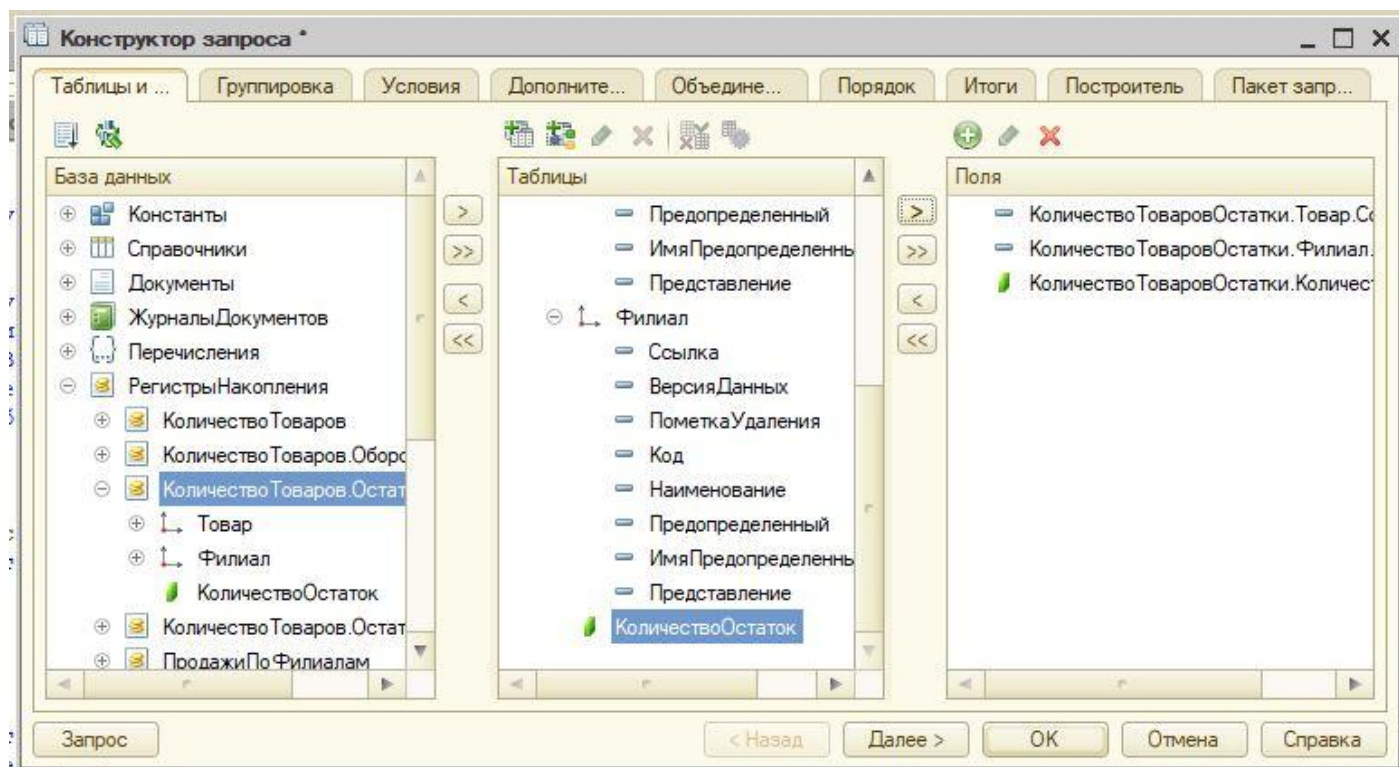
```
□/}_{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
КонецПроцедуры
```

Получение данных через запрос. [Интересный набор статей про регистр накоплений.](#)

Регистр накопления с видом "Остатки" позволяет использовать виртуальные таблицы "Обороты" и "Остатки".

Используем [Конструктор запроса](#). Так как нужно просто получить остатки на указанную дату, выбираем регистр, таблицу и поля



Итоговая процедура:

```
&НаСервереБезКонтекста
```

```
Процедура ВзятьОстаткиНаСервере(НужнаяДата)
```

```
□Запрос = Новый Запрос;
```

```
□Запрос.Текст = "ВЫБРАТЬ
```

```
□□КоличествоТоваровОстатки.Товар.Ссылка.Описание как Товар,
```

```
□□КоличествоТоваровОстатки.Филиал.Ссылка.Наименование как Филиал,
```

```
□□КоличествоТоваровОстатки.КоличествоОстаток как Колво
```

```
□□ИЗ
```

```
□□РегистрНакопления.КоличествоТоваров.Остатки(&Дата) КАК КоличествоТоваровОстатки";
```

```
□Запрос.установитьпараметр("Дата",НужнаяДата);
```

```
□Результат = Запрос.Выполнить();
```

```
□Если не Результат.Пустой() Тогда
```

```
□□ВыборкаДетальныеЗаписи = Результат.Выбрать();
```

```
□□Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
```

```

□□Сообщить(ВыборкаДетальныеЗаписи.Филиал + ВыборкаДетальныеЗаписи.Товар +
ВыборкаДетальныеЗаписи.Колво);
□КонецЦикла;
    КонецЕсли;
КонецПроцедуры

```

&НаКлиенте

Процедура ВзятьОстатки(Команда)

□Если не ЗначениеЗаполнено(ДатаФормированияОстатков) тогда

□ДатаФормированияОстатков = ТекущаяДата();

□КонецЕсли;

□ВзятьОстаткиНаСервере(ДатаФормированияОстатков);

КонецПроцедуры

Получение данных через менеджер.

&НаСервереБезКонтекста

Процедура ВзятьОстаткиНаСервере2()

□РегМенеджер = РегистрыНакопления.КоличествоТоваров;

□Остатки = РегМенеджер.Остатки();

□Для каждого ТекОст из Остатки Цикл

□□Сообщить(Строка(ТекОст.Филиал) + " " + Строка(ТекОст.Товар) + ": " + ТекОст.Количество);

□КонецЦикла;

КонецПроцедуры

Значения в возвращенной таблице являются ссылками, из которых еще нужно получить значения. А это дополнительные запросы к базе. В примере прокатило, т к в этих справочниках основное представление в виде наименование. Если изменить в справочнике на в виде кода, то данный пример выведет числа вместо наименований филиалов и товаров. Т е если в базе 10 филиалов, 10 товаров и представление в виде кода, то будет генериться 100 запросов к базе просто для получения имени. А это очень неэффективно. Так что этот способ использовать осознанно.

Регистры сведений

??????? ?????????????? ??? ?????????? ??????? (?????????) ? ?????????????????????? ????? (? ?????????
 ??????????) ? ?????????????????? ?????????????????? ?????????????? (????????????). ?????????????????? ??????????
 ?????????????? ?????????????? ???????, ?????? ?????????? ?????????????? ?????? ??????????????. ?????? ??????? ??????
 ?????? ?????????????? ??? ?????????????? (?????????? ??????????????????). ?????????????????? ?????????? ?????????????????,
 ?.?. ?????????, ? ?????????? ?????????????? ?????????? ??????????, ?????? (????? ?????????? ??????????????????) ?
 ?????????????? (????? ?????????? ?????????? ??????????????????).

????????????????????? ?????????????? ?????????? ?????????? ?????????????? ?? ?????????????? ??????? ??????????????
 ?????????????????? ?

????? ?????????????? ?????? ?????????????????, ?? ?????? ?????????????? ??????????????.

????????????? ? ?????????? ??????????. ?????? ?????? ??????????????????:

- ?????????? ?????????? ?????????????????;
- ?????????? ?????????? ?? ??????????, ?????? (????? ?????????????????) ? ?????????????? (????? ?????????? ?????????????); ?? ?????????? ?????? ?????????? ?????? ? ?????????????? ? ?????????? ? ?????? ?????????? ?? ?????????? ?????? ??????.
- ?????????????? ? ?????????????? ?????????? ?????? ?????????;
- ??????? ??????? ?????????.

```
// Добавление записи в независимый непериодический регистр сведений
НаборЗаписей = РегистрыСведений.ВерсииПодсистем.СоздатьНаборЗаписей(); // Этап 1
НаборЗаписей.Отбор.ИмяПодсистемы.Установить(ИмяПодсистемы); // Этап 2
// Этап 3
НоваяЗапись = НаборЗаписей.Добавить();
НоваяЗапись.ИмяПодсистемы = ИмяПодсистемы;
НоваяЗапись.Версия = НомерВерсии;
НаборЗаписей.Записать(); // Этап 4
```

????? ?? ?????????? ??????, ?? ?????????????????? ?????? ??????????. ? ??????? ? ?????????, ?????????????????? ?? ?????? ??????????, ?????????? ?????? ?????????.

??????????, ?????? ?????????, ? ?????????? ??????????? ? ?????????? - ???????????, ??????????? - ??????. ?????? ?????? ?????????:

МояСтрока	МоеЧисло	Результат
олдж	7	67
олдж	8	67
фыва	4	67

????? ?????? ?????????????? ?????? ?????? ?? ??????????? ?????????? "?????" ? ?????? ?????????? ?????? ??????, ?? ? ?????????? ?????????? 2 ?????? (????? ? ?????). ??????? ?????????????????? ? ?????? ??????? ???????????, ?????????? ?? ?????????? ??? ??????? ????????????. ? ? ?????? ?????? ?????????? ?????? ?????, 7, ? ? ?????? ??????? ?????? ?????????? ?????????? ?????? 8, ?????? ??????. ?? ?????? ??????? ?????? ?????? ?????, ?? ?????? ?????????????????? ??? ?????????? ??????????.

```
// Добавление записи в независимый периодический регистр сведений
НаборЗаписей = РегистрыСведений.КурсыВалют.СоздатьНаборЗаписей(); // Этап 1
// Этап 2
НаборЗаписей.Отбор.Валюта.Установить(Доллар);
НаборЗаписей.Отбор.Период.Установить(НачалоДня(ТекущаяДата()));
// Этап3
НоваяЗапись = НаборЗаписей.Добавить();
НоваяЗапись.Период = ТекущаяДата();
```

```
НоваяЗапись.Валюта = Доллар;  
НоваяЗапись.Курс = 57.92;  
НоваяЗапись.Кратность = 1;  
НаборЗаписей.Записать(); // Этап 4
```

????????? ??????. ??? ?????? ??????? ?????????? ?????????? ?????????? ??? ?????????? ??????????????
?????????. ?????? ?????????????????? ???????.

```
// Редактирование записей с использованием объекта НаборЗаписей  
НаборЗаписей = РегистрыСведений.КурсыВалют.СоздатьНаборЗаписей(); // Этап 1  
// Этап 2  
НаборЗаписей.Отбор.Период.Установить(ДатаКурса);  
НаборЗаписей.Отбор.Валюта.Установить(Доллар);  
НаборЗаписей.Прочитать(); // Этап 3  
Для Каждого Запись Из НаборЗаписей Цикл  
    Запись.Курс = 57.84; // Этап 4  
КонецЦикла;  
НаборЗаписей.Записать(); // Этап 5
```

?????? ???????. ??????????? ?????????? ?????? ??????.

```
Запрос = Новый Запрос;  
Запрос.Текст =  
    "ВЫБРАТЬ  
    | КурсыВалют.Период,  
    | КурсыВалют.Валюта,  
    | КурсыВалют.Курс  
    |ИЗ  
    | РегистрСведений.КурсыВалют КАК КурсыВалют";  
  
Выборка = Запрос.Выполнить().Выбрать();  
Пока Выборка.Следующий() Цикл  
    // обход результата выполнения запроса  
КонецЦикла;
```

??? ?????????????? ?????????? ??? ?????????????? ??????? ?????????????????? (?????? ????????? ??????
?????????? ?????) ? ????????????? (?????? ????????? ??????? ??????????? ?????).


```

// Получение записи, у которой валюта равна значению из переменной «ВыбраннаяВалюта»
// и период БОЛЬШЕ или равен значению из переменной «ВыбраннаяДата»
Запрос = Новый Запрос;
Запрос.Текст =
    "ВЫБРАТЬ
      | КурсыВалютСрезПервых.Период,
      | КурсыВалютСрезПервых.Валюта,
      | КурсыВалютСрезПервых.Курс
    |ИЗ
      | РегистрСведений.КурсыВалют.СрезПервых(&Период, Валюта = &Валюта) КАК
КурсыВалютСрезПервых";

Запрос.УстановитьПараметр("Валюта", ВыбраннаяВалюта);
Запрос.УстановитьПараметр("Период", ВыбраннаяДата);

Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    // обход результата выполнения запроса
КонецЦикла;

```

???? ????? ??????? ?? ??????? ? ?????? ?? ??????? ??? ??????????? ????? ??? ??????????
 ??????????????:

```

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

```

```

// Получить текущую строку табличной части.
СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;
// Установить цену.
СтрокаТабличнойЧасти.Цена = РаботаСоСправочниками.РозничнаяЦена(
    Объект.Дата, СтрокаТабличнойЧасти.Номенклатура);
// Пересчитать сумму строки
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти)

```

Проведение одним документом в нескольких регистрах через конструктор

Справа добавить регистры и настроить сопоставление полей.

Revision #13

Created 16 January 2025 15:06:30 by Admin

Updated 28 January 2025 14:03:53 by Admin