

Класс Справочники

У всех справочников есть стандартные реквизиты: Код и Наименование. Справочник обновляется в момент обращения или изменения со стороны пользователя, при изменении другим пользователем или программным изменением автоматического обновления нет, данные кэшируются, поэтому нужно вручную обновить справочник.

Существует тип данных Составной реквизит, создается при помощи троеточия в поле настройки типа данных в свойстве реквизита.

Настройка внешнего вида

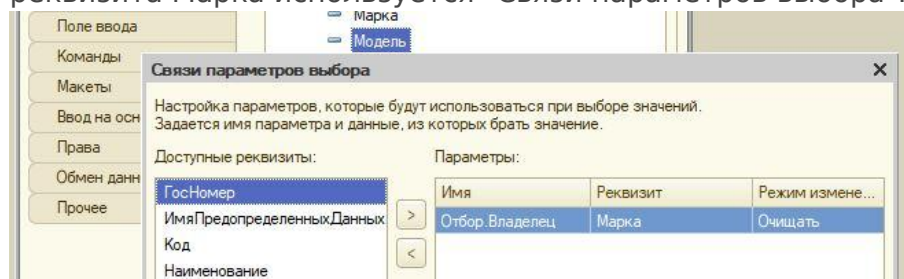
У реквизита есть свойство Использование. При создании иерархических справочников можно указать, для чего использовать этот реквизит. При использовании "Для элемента" этого реквизита не будет у группы.

Предопределенные элементы. ПКМ на справочнике -> Открыть предопределенные данные. Используется в частности для настройки значений по умолчанию в Документах, ... Для запрета удаления предопределённых элементов нужно выключить во всех ролях следующие права (по умолчанию выключены):

- «ИнтерактивноеУдалениеПредопределённыхДанных»
- «ИнтерактивнаяПометкаУдаленияПредопределённыхДанных»
- «ИнтерактивноеСнятиеПометкиУдаленияПредопределённыхДанных»
- «ИнтерактивноеУдалениеПомеченныхПредопределённыхДанных».

Подчиненные справочники. В свойствах справочника -> Владелец добавить справочник-владелец. Может быть несколько справочников-владельцев, но владелец конкретного элемента подчиненного справочника может быть один элемент одного из справочников-владельцев. Множественное владение не поддерживается. [Детали удаления элемента родительского справочника](#)

Связь параметров выбора. Предположим, необходимо хранить список наших автомобилей. Среди параметров в частности есть модель, марка, и т.д. (гос. номер, ...). У одной модели может быть несколько марок. Логично создать справочник Модель и подчиненный справочник Марка. Однако если просто в справочнике Автомобили в реквизитах указать ссылки на Модель и Марка, то при поиске марки будут отображаться все марки, а не только для выбранной модели. Для исправления этого поведения в свойствах реквизита Марка используется "Связи параметров выбора".



Табличные части. Используется для отражения информации, связанной с данным элементом, но не имеющей связанной объектной сущности. Например, для справочника "Товары" может быть создана табличная часть "ЕдиницыИзмерения".

Выбор между подчиненным справочником и табличной частью.

Это близкие элементы, поэтому нужно хорошо подумать перед выбором.

- При использовании табличной части нельзя ссылаться на строки табличной части, т.е. нельзя будет создать реквизит, соответствующий понятию «строка табличной части».
- Если в перспективе может возникнуть потребность ссылаться на подчиненные объекты, например, создавать ссылающиеся на них реквизиты документов или других справочников, то лучше сразу завести подчиненный справочник. Если же ссылка на такие сведения не имеет смысла и никогда не может быть типом какого-либо реквизита, тогда можно завести табличную часть.

Пример ситуаций и выбор между подчиненным справочником и табличной частью.

Ситуация 1 В базе данных необходимо хранить список расчетных счетов каждого контрагента. Почти наверняка в платежных документах будет необходимо, кроме контрагента, указывать его расчетный счет. Такая информация имеет объектную связь и может быть идентифицирована как "расчетный счет". Возможное решение: Справочник "Контрагенты" и подчиненный ему справочник "РасчетныеСчета" с полями "Банк", "Номер", "КоррСчет".

Ситуация 2 Для справочника "Номенклатура" нужно хранить список единиц измерения для каждого товара с указанием коэффициента пересчета в основную единицу измерения. Эти сведения подбираются из справочника "ЕдиницыИзмерения", который хранит все существующие в природе единицы измерений. Каждая строка такого подчиненного списка не имеет собственной объектной сущности, а нужна только для пересчета из одной единицы измерения в основную. Возможное решение: Справочник "Номенклатура" и табличная часть "ЕдиницыИзмерения" с полями "ЕдиницаИзмерения" и

"КоэффициентПересчета".

Ситуация 3 Допустим, встроенной системы задания прав пользователей не хватает для некоторых специальных приложений. Например, часто требуется разработать механизм утверждения документов разными пользователями. Тогда для сотрудника нужно хранить список документов, которые он может утверждать, и с этим отлично справится табличная часть "УтверждаемыеДокументы" справочника "Сотрудники". Такая информация не имеет объектной связи, а просто связывает документ и сотрудника, поэтому вряд ли понадобится когда-либо в будущем создавать ссылки на нее. Возможное решение: Справочник "Сотрудники" и табличная часть "УтверждаемыеДокументы" с полем "Документ" и флажком "Утверждается". Заметим, что эту задачу также можно решить с использованием регистров сведений.

Ситуация 4 Для справочника "Сотрудники" требуется хранить сведения о составе семьи сотрудника, т.е. вносить информацию о членах семьи и их родственных отношениях к сотруднику (муж, жена, сын, дочь и т.д.). Обычно эта списковая информация полностью подчинена элементу справочника "Сотрудники", и возникает мысль о том, чтобы завести табличную часть. Но если подумать, то такая информация имеет четкую объектную природу и может быть идентифицирована как "член семьи". Для некоторых приложений может потребоваться создавать ссылки на членов семьи сотрудника. Аналогичная ситуация наблюдается со сведениями об образовании и о предыдущих местах работы сотрудника. Возможное решение: Выбор между подчиненным справочником и табличной частью зависит от назначения конфигурации. Если в будущем может возникнуть потребность создавать ссылки на такие сведения, то лучше завести подчиненные справочники "СоставСемьи", "Образование" и "ТрудоваяДеятельность".

Важное замечание. Необходимо помнить, что при обращении к элементу справочника он весь, вместе со всеми табличными частями, считывается из базы данных в память. Если табличная часть содержит достаточно большое количество строк это может заметно сказаться на производительности системы.

Таким образом, табличную часть стоит использовать, если не надо хранить ссылки на элементы, и количество элементов ограничено.

Программная работа с элементами справочника

Подборка примеров кода [ссылка](#)

Поиск по реквизиту. Код обработки, есть справочник Цвета.

```
&НаКлиенте  
Процедура ПолучитьОбъектПоНаименованию(Команда)  
□ТекстНаименования = НайтиОбъект(Объект.Наименование);  
□Сообщить(ТекстНаименования);  
КонецПроцедуры  
  
&НаСервереБезКонтекста
```

Функция НайтиОбъект(Наименование)

□ЦвСсылка = Справочники.Цвета.НайтиПоНаименованию(Наименование);

□Если ЦвСсылка = Справочники.Цвета.ПустаяСсылка() Тогда

□□Возврат("Цвет не найден");

□Иначе

□□Возврат(ЦвСсылка.Наименование + ": " + ЦвСсылка.Код);

□КонецЕсли;

КонецФункции

P.s. На клиент нельзя вернуть объект. Можно при желании вернуть структуру, или например ссылку, которую можно использовать в других подпрограммах:

&НаКлиенте

Процедура ПолучитьОбъектПоНаименованию(Команда)

□СсылкаНаЦвет = НайтиОбъект(Объект.Наименование);

□Если СсылкаНаЦвет = Неопределено Тогда

□□Сообщить("Цвет не найден");

□Иначе

□□ВывестиПоСсылке(СсылкаНаЦвет);

□КонецЕсли;

КонецПроцедуры

&НаСервереБезКонтекста

Функция НайтиОбъект(Наименование)

□ЦвСсылка = Справочники.Цвета.НайтиПоНаименованию(Наименование);

□Если ЦвСсылка = Справочники.Цвета.ПустаяСсылка() Тогда

□□Возврат(Неопределено);

□Иначе

□□Возврат(ЦвСсылка);

□КонецЕсли;

КонецФункции

&НаСервереБезКонтекста

Процедура ВывестиПоСсылке(ЦветСсылка)

□Сообщить(ЦветСсылка.Код);

КонецПроцедуры

Создание групп или элементов.

НовыйЭлемент = Справочники.Номенклатура.СоздатьЭлемент();

НовыйЭлемент.Реквизит1 = "КакоеТоЗначение";

```
НовыйЭлемент.Записать();
```

Выборка элементов

```
Выборка = Справочники.Номенклатура.Выбрать();
```

```
Пока Выборка.Следующий() Цикл
```

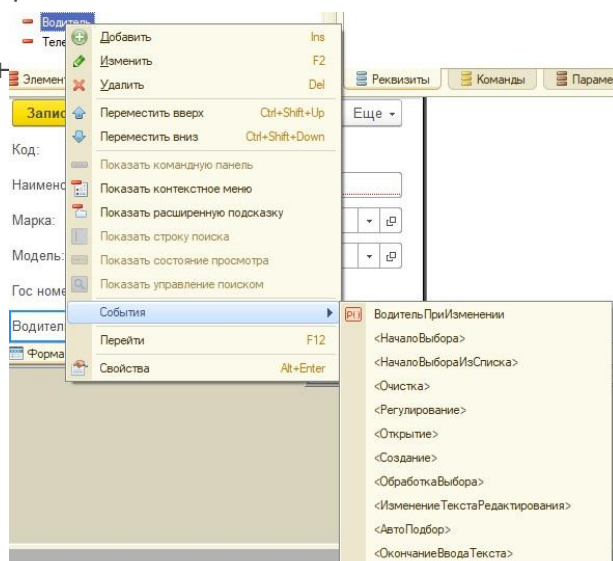
```
ТекСсылка = Выборка.Ссылка;
```

```
Сообщить(ТекСсылка.Наименование);
```

```
КонецЦикла;
```

Автоматическое заполнение реквизитов. Например есть справочник Сотрудники, у которого есть реквизит Рабочий телефон. Есть справочник Автомобили, у которого есть реквизит Водитель и Рабочий телефон. Нужно, чтобы при выборе водителя, происходило заполнение рабочего телефона.

Вариант 1. В редактировании



События - ПриИзменении

&НаКлиенте

Процедура ВодительПриИзменении(Элемент)

```
□Объект.ТелефонВодителя = ПолучитьТелефон(Объект.Водитель);
```

КонецПроцедуры

&НаСервереБезКонтекста

Функция ПолучитьТелефон(СсылкаНаВодителя);

```
□Возврат СсылкаНаВодителя.РабочийТелефон;
```

КонецФункции

Однако, при изменении в справочнике Сотрудники внутреннего телефона, обновление не происходит.

Вариант 2. Обновление будет при создании связанных справочников Сотрудники - ТелефоныСотрудников и т. д. Обработчик, который добавляет первый указанный телефон пользователя

&НаКлиенте

Процедура ВодительПриИзменении(Элемент)

□Перем Телефон;

□НайтиТелефоныВодителя(Объект.Водитель, Телефон);

□Если Телефон <> Неопределено Тогда

□□Объект.ТелефонВодителя = Телефон;

□КонецЕсли;

КонецПроцедуры

&НаСервереБезКонтекста

Процедура НайтиТелефоныВодителя(СсылкаНаСотрудника, СсылкаНаТелефон)

□Запрос = Новый Запрос(

 "ВЫБРАТЬ

 | Телефоны.Ссылка как ТелефонВодителя

 |ИЗ

□□ | Справочник.ТелефоныСотрудников КАК Телефоны

 |ГДЕ

 | Телефоны.Владелец = &Ссылка");

 Запрос.УстановитьПараметр("Ссылка", СсылкаНаСотрудника);

□Результат = Запрос.Выполнить();

□Если не Результат.Пустой() Тогда

□□ВыборкаДетальныеЗаписи = Результат.Выбрать();

□□ВыборкаДетальныеЗаписи.Следующий();

□□СсылкаНаТелефон = ВыборкаДетальныеЗаписи.ТелефонВодителя.Ссылка;

 КонецЕсли;

КонецПроцедуры

Можно доработать процедуру, возвращая массив ссылок, на клиенте сохраняя первую и сообщая остальные в виде всплывающего сообщения.

Есть косяк: При изменении владельца номера телефона, в справочнике Автомобили остается старый Водитель и не меняется ссылка на номер телефона. Возможно, необходима обработка При изменении на ТелефоныСотрудников. Типа уведомления о том, что нужно обновить данные в таком-то справочнике для такой-то записи.

Обработка события удаления элементов

Справочник Менеджеры подчинен справочнику Прайс-лист. В модуле объекта справочника Прайс-лист обработка удаления элемента.

Процедура ПередУдалением(Отказ)

□НайтиПодчиненныеЭлементы(ЭтотОбъект.Ссылка, Отказ);

КонецПроцедуры

&НаСервере

Процедура НайтиПодчиненныеЭлементы(СсылкаНаОбъект, Отказ)

Запрос = Новый Запрос(

"ВЫБРАТЬ

| Менеджеры.Ссылка.Наименование как ИмяМенеджера

|ИЗ

□□ | Справочник." + Метаданные.Справочники["Менеджеры"].Имя + " КАК Менеджеры

|ГДЕ

| Менеджеры.Владелец = &Ссылка");

Запрос.УстановитьПараметр("Ссылка", СсылкаНаОбъект);

□Результат = Запрос.Выполнить();

□Если Результат.Пустой() Тогда

□□Отказ = Ложь;

□Иначе

□□Сообщить("Нельзя удалить элемент, так как он владелец элементов.");

□□ВыборкаДетальныеЗаписи = Результат.Выбрать();

□□СписокЗависимостей = "";

□□Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

□□□СписокЗависимостей = СписокЗависимостей + " " + ВыборкаДетальныеЗаписи.ИмяМенеджера;

□□КонецЦикла;

□□Сообщить(СписокЗависимостей);

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

Группа или элемент

Метод «ЭтоГруппа()» возвращает 1 в случае группы и 0 в случае элемента.

Доступ к табличной части

&НаСервереБезКонтекста

Процедура ПолучитьТабличнуюЧастьНаСервере(КанцелярияСсылка)

□Набор = КанцелярияСсылка.СоставНабора;

```
□Если Набор.Количество() > 0 Тогда
□□Сообщить("Состав набора:");
□Иначе
□□Сообщить(КанцелярияСсылка.Наименование + " не набор.");
□КонецЕсли;
□Для каждого СтрокаТаблицы из Набор Цикл
□□Сообщить(СтрокаТаблицы.Канцелярия.Наименование + ", " + СтрокаТаблицы.Количество + " шт.");
□КонецЦикла;
КонецПроцедуры
```

Revision #26

Created 8 January 2025 17:30:21 by Admin

Updated 12 February 2025 17:07:52 by Admin