

# Техническое описание продукта или возможности расширенного конфигурирования.

## Оглавление

Средства разработки.....	3
Клиент-сервер для больших и маленьких.....	3
База данных MS SQL .....	4
Основные объекты системы Айтида.....	4
Документы .....	5
Общие обязательные реквизиты, относящиеся ко всем документам: .....	5
Реквизиты, относящиеся к документам, отражающим изменения во взаиморасчетах с контрагентами: .....	6
Реквизиты, относящиеся к документам, отражающим изменения в складском учете:.....	6
Типы документов.....	8
Дополнительные таблицы документов.....	9
Номера партий товаров (barcodes2). .....	12
Дополнительные реквизиты документов, определенные в точке зрения (docsanalit_ex). .....	13
Дополнительные файлы, прикрепленные к документу (spec000_img).....	14
История изменения статусов документов (statuslist).....	14
Список моделей учета, на которые должен влиять документ (accountlist).....	15
Таблица связей между документами (crosslink ). .....	15
Пример заполнения таблиц документа. ....	18
Серийные номера.....	21
Журнал всех документов. ....	22
Дополнительные реквизиты документов, определенные в точке зрения. ....	23
Таблица связей между документами. ....	27
Справочники .....	30
Типы справочников системы Айтида.....	31
Дополнительные реквизиты справочников.....	35
Дополнительные справочники, определенные при настройке системы. ....	36
Таблица sprmain. ....	37
Иерархия в справочниках. ....	38
Справочник аналитик. ....	39
Характеристики товаров. ....	40
Пример заполнения таблиц справочника.....	41
Регистры учета. ....	55

Основные регистры системы:.....	55
Общие принципы хранения информации в регистрах. ....	56
Регистр учета движений ТМЦ.....	58
Регистр учета резервов ТМЦ. ....	66
Регистр учета денежных средств. ....	70
Регистр учета взаиморасчетов с контрагентами. ....	74
Регистр учета взаиморасчетов между филиалами.....	79
Регистр учета номеров партий ТМЦ. ....	81
Регистр учета оборотной тары. ....	85
Регистр учета цен реализации.....	87
Регистр учета серийных номеров. ....	91
Регистр учета проводок (журнал хозяйственных операций).....	94
Регистр учета расчетов с сотрудниками организаций. ....	100
Регистр учета заказов поставщикам. ....	102
Регистр учета бюджетных показателей.....	105
Универсальный регистр учета. ....	108
Регистр учета операций с объектами основных средств и нематериальных активов. ....	111
Регистр учета операций с ценными бумагами. ....	117
Регистр учета калькуляций и состава объектов ОС. ....	119

# 1. Архитектура решений

## Средства разработки

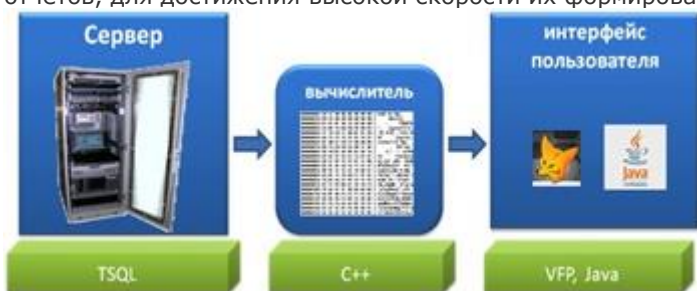
Программные продукты «Айтида» разрабатываются с использованием современных средств разработки – языков программирования **C++**, **Java**, **MS TSQL** и **MS VFP**.

Почему мы используем несколько языков?

Напомним, что программные продукты Айтида являются **клиент-серверными приложениями**. Серверная часть системы разработана на MS TSQL – наиболее известном и распространенном языке программирования, предназначенном для решения именно этих задач. TSQL позволяет пользователю осуществлять быстрые выборки информации из базы данных по заданным фильтрам, группировать данные, сортировать их.

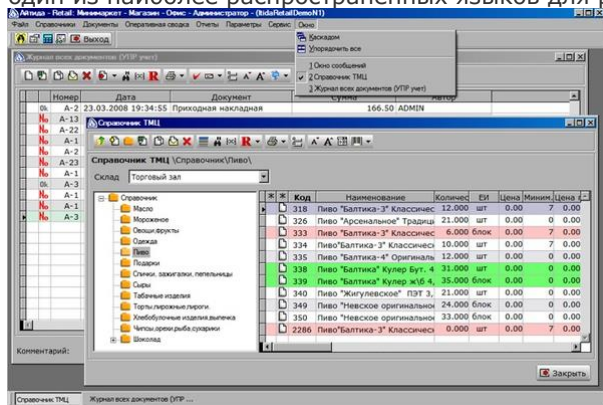
После того, как по запросу пользователя необходимая информация извлечена из базы данных (с помощью TSQL), как правило, требуется произвести какие-либо расчеты. За расчеты в наших программных продуктах отвечает центральный вычислитель системы.

Центральный вычислитель разработан на языке C++. Этот язык программирования является одним из «самых быстрых» и применяется обычно в программах, предназначенных для произведения большого количества сложных математических расчетов. Центральный вычислитель используется и генератором отчетов, для достижения высокой скорости их формирования.



После того, как нужная информация выбрана из базы данных и обработана вычислителем, ее можно представить пользователю. Разработка пользовательского интерфейса сейчас ведется с применением двух языков программирования:

MS VFP используется для разработки стандартного функционала системы, связанного с текущей работой пользователей – работа справочниками и документами, журналами и проводками, обмен с оборудованием и общая настройка системы. VFP выбран для решения этой задачи исторически, как один из наиболее распространенных языков для разработки пользовательских интерфейсов.



Использование данного языка позволяет не только быстро разрабатывать новые элементы системы и описывать их алгоритмы, но и обеспечить удобство и скорость работы пользователей.

## Клиент-сервер для больших и маленьких

**Клиент-сервер** (англ. *Client-server*) — сетевая архитектура, в которой устройства являются либо клиентами, либо серверами. Клиентом является запрашивающая машина (обычно ПК пользователя), сервером - машина, которая отвечает на запрос.

Действительно, программный продукт, разработанный с применением технологии клиент-сервер это фактически две программы. Одна из них работает на компьютере пользователя (отображает информацию, позволяет сформировать условия выборки данных и передать их на сервер), другая работает на сервере, принимает запросы клиентских станций, осуществляет выбор запрошенных данных, и возвращает их пользователю.

Такая технология работы программы значительно сложнее обычной, но обеспечивает максимальную производительность. При работе более простых систем именно пользовательская станция осуществляет

выборку нужных данных. Поэтому при любом запросе (например, при формировании отчета или, при открытии журнала документов) по сети передается большое количество лишней информации - замедляя работу всех пользователей, излишне нагружая сеть.

К основным достоинствам технологии клиент-сервер относятся:

- Более высокая производительность
- Низкие требования к рабочим станциям
- Низкие требования к пропускной способности сети

И кроме того:

- Надежность хранения данных
- Защита от несанкционированного доступа

Для работы клиент-серверного приложения не обязательно наличие выделенного сервера. И клиентскую и серверную часть программы можно установить на компьютер пользователя. При этом все достоинства технологии сохраняются, а минимальные требования к рабочей станции достаточно лояльны:

- Pentium III
- 512 Mb RAM

Следует отметить, что для работы программных продуктов Айтида приобретение платной лицензии MS SQL сервер не обязательно. Для большинства компаний на долгое время будет достаточно бесплатной версии MS SQL Express, которая поставляется в составе нашего дистрибутива. Подробно об этом продукте можно прочитать здесь: <http://www.microsoft.com/express/ru/sql/default.aspx>

## База данных MS SQL

Microsoft SQL Server 2008 Express Edition (SQL Server Express) – это мощная и надежная программа по управлению данными, обеспечивающая множество функций, защиту данных и высокую производительность для встроенных приложений-клиентов, "легких" веб-приложений и локальных хранилищ данных. SQL Server Express предназначен для упрощенного развертывания и быстрого создания прототипов; его можно получить бесплатно и свободно распространять в составе приложений. SQL Server Express естественным образом интегрируется с другими продуктами, входящими в серверную инфраструктуру.

Основным достоинством наличия единой базы данных является то, что она сама, через внутренние связи, триггеры и хранимые процедуры, сохраняет и поддерживает целостность хранимых данных. Таким образом, не зависимо от того какое клиентское приложение используется для внешнего управления данными, все внутренние взаимосвязанные объекты базы (исходные данные документов и справочников и производные от них регистры и расчеты) будут находиться в актуальном состоянии.

## Основные объекты системы Айтида

Все объекты системы Айтида делятся на следующие группы:

1. Объекты, содержащие исходные данные вводимы пользователем – документы и справочники. Эти объекты имеют интерфейсы для пользователя для ввода, корректировки и удаления введенных данных.
2. Объекты, информация в которые поступает автоматически, в результате производимых расчетов над документами и справочниками – регистры учета. Информация в регистрах не

доступна для непосредственного изменения пользователями системы и автоматически поддерживается в актуальном состоянии средствами базы данных.

3. Объекты, содержащие информацию о настройках системы и прочую служебную информацию.

## Документы

Документ – это средство для ввода, хранения и отображения информации о совершаемых хозяйственных действиях, которые влекут за собой движения и изменения ресурсов. При помощи документов организуется ввод информации в систему, а также ее просмотр и, если необходимо, корректировка. В большинстве своем документы являются электронными аналогами стандартных бумажных документов. Структура каждого конкретного документа определяется его назначением в системе. У любого вида документов существует три обязательных реквизита, которые, в случае необходимости, создаются автоматически - номер, дата и автор документа.

В зависимости от типа фиксируемых операций, документы делятся на разные типы и содержат различные реквизиты. Часть реквизитов является обязательной для всех типов документов и служит для идентификации самого документа, идентификации лица создавшего или изменившего документ и т.п. Другая часть реквизитов является специфической для конкретного типа документов и содержит специфическую для данного типа информацию. В системе Айтида применяется единое правило именования реквизитов в структуре таблиц объектов. Если реквизиты разных объектов содержат однородную информацию, например, код контрагента, то имя и тип поля в соответствующих таблицах будут совпадать. Ниже приведены основные группы используемых реквизитов.

### Общие обязательные реквизиты, относящиеся ко всем документам:

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
Идентификатор документа.	identity_column int	Является уникальным идентификатором документа, по которому производится любое обращение к документу. Идентификатор уникален внутри одного типа документов.
Номер документа	ndok char(20)	Не уникальный номер документа в системе. Может быть произвольно установлен пользователем. Служит для идентификации документа в печатных формах и отчетах.
Дата документа	date datetime	Дата документа. Вводится пользователем. Служит вместе с номером документа для идентификации документа в печатных формах и отчетах. Является основным источником даты для отражения в учете.
Организация (фирма)	firm char(7)	Код организации, от имени которой выписывается документ.
Расчетный счет	rs char(20)	Расчетный счет организации.
Комментарий	note varchar(250)	Комментарии к документу, вводимые пользователем.
Валюта документа	cur char(3)	Код валюты, в которой выписан документ. Документы могут быть выписаны в любой валюте. В момент отражения документа в учете все указанные в документе суммы будут пересчитаны в валюту учета системы по курсу документа. Для корректного отражения документа в учете, указанная валюта должна быть заведена в

		справочнике валют и должен быть указан курс конвертации такой валюты к валюте учета на дату документа.
Курс валюты документа	cur_rat float	Курс, по которому будет произведен пересчет сумм документа в валюту учета. Может быть введен вручную пользователем, а может быть взят из справочника курсов валют на дату документа.
Филиал документа	branch char(3)	Код из справочника филиалов, к которому имеет отношение документ. В большинстве случаев, документ будет находить отражение в учете только в филиале, к которому он имеет отношение.
Филиал, получатель документа	branch_d char(3)	Филиал, в который необходимо передать документ. Например, документ внутреннего перемещения ТМЦ содержит код филиала получателя ТМЦ.
Автор документа	author varchar(128)	Код из справочника пользователей автора документа. Автором считается пользователь, который первым добавил документ в систему. Пользователи, осуществляющие дальнейшие операции с документом отражаются в истории работы с документом и не влияют на значение поля автор.

**Реквизиты, относящиеся к документам, отражающим изменения во взаиморасчетах с контрагентами:**

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
Код контрагента.	client char(7)	Код контрагента из справочника контрагентов. В случае, если данный реквизит в документе не заполнен, то документ не будет участвовать в учете взаиморасчетов с контрагентами. Так же, если карточка указанного контрагента в своем реквизите Филиал содержит филиал, отличный от рабочего, то документ не будет участвовать в учете взаиморасчетов с контрагентами.
Код договора	contract char(10)	Код из справочника договоров с контрагентом. Значение поля может использоваться как один из разрезов ведения учета взаиморасчетов с контрагентами.
Сумма	summa float	Сумма документа в валюте документа. Основная сумма, используемая для учета взаиморасчетов с контрагентами. Документ может содержать много различных сумм. В учете взаиморасчетов с контрагентами участвует сумма, находящаяся на одном «уровне» с контрагентом. Если контрагент указывается в шапке документа, то сумма используется из шапки документа. Если контрагент указан в многострочной части документа, то и сумма берется оттуда же.

**Реквизиты, относящиеся к документам, отражающим изменения в складском учете:**

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
Склад.	sklad char(3)	Код склада из справочника складов. В случае, если

		данный реквизит в документе не заполнен, то документ не будет участвовать в складском. Так же, если карточка указанного склада в своем реквизите Филиал содержит филиал, отличный от рабочего, то документ не будет участвовать в складском учете.
Материально ответственное лицо	mol char(7)	Код МОЛ из справочника сотрудников. Значение поля может использоваться как один из разрезов ведения складского учета.
Товар	nn char(10)	Код из справочника товаров. Является основным объектом складского учета. Чаще всего, данный реквизит находится в многострочной части документа.
Цена	cena float	Цена товара, участвующего в операции. Назначение данного реквизита может изменяться в зависимости от типа документа. Цена товара используется, только если указано количество товара, участвующего в операции. Если количество товара не указано, или указано 0, то вместо цены используется реквизит Сумма товара. Цена указывается в валюте документа, поэтому в момент принятия документа к учету производится пересчет указанной цены
Сумма	summa float	Сумма товара, участвующего в операции. Назначение данного реквизита может изменяться в зависимости от типа документа. В складском учете участвует сумма, находящаяся на одном «уровне» с товаром. Если товар указывается в многострочной части документа, то сумма используется из многострочной части документа.
Количество	kolp float	Количество товара участвующего в операции. Назначение данного реквизита может изменяться в зависимости от типа документа.
Единица измерения	ed char(15)	Код из справочника единиц измерения. Определяет в каких единицах измеряется указанное количество товара.
Коэффициент пересчета количества	koef_e float	Коэффициент пересчета количества, указанного в документе в количество в основной единице измерения товара, указанной в карточке товара. Складской учет товаров ведется в основной единице товаров, которая указана в карточке товара, поэтому для принятия документа к складскому учету система производит пересчет количества, указанного в документе в основную единицу измерения. Формула для пересчета: kolp*koef_e
Коэффициент пересчета цены	koef_c float	Коэффициент пересчета указанной цены товара, в цену товара в основной единице измерения, указанной в карточке товара.
Код списка налогов	kodn char(2)	Код из справочника списков налогов. Используется для расчета налогов, взимаемых с операции.

**Примечание.** Сумма и цена товара, которая в итоге принимается к учету (при оприходовании и списании товара) дополнительно зависит от параметров учета НДС, указанных в карточке окружения для конкретной организации и модели учета. В итоге, конечная формула расчета количества, цены и суммы товара, принимаемого к учету, выглядит следующим образом:

*Сумма = ЕСЛИ(Количество == 0, ЕСЛИ( УЧЕТБЕЗНДС, СуммаБезНДС, Сумма), Количество \* ЕСЛИ( УЧЕТБЕЗНДС, ЦенаБезНДС, Цена) / ЕСЛИ(ВалютаДокумента == ВалютаУчета, 1, КурсДокумента)*

$Цена = Цена / ЕСЛИ( ВалютаДокумента == ВалютаУчета, 1, КурсДокумента) * КоэффициентПересчетаЦены$

$Количество = Количество * КоэффициентПересчетаКоличества$

### Типы документов.

Все документы в системе Айтида делятся на типы, в зависимости от выполняемых операций. Документы одного типа имеют одинаковый набор базовых реквизитов. Кроме того, для каждого типа документов в точке зрения могут быть определены дополнительные реквизиты, необходимые для организации конкретных схем учета. Каждый тип документа в системе имеет свой уникальный код. В следующей таблице приведены типы документов, коды и таблицы, в которых размещаются базовые реквизиты документов.

Код	Наименование типа документа	Таблица с реквизитами шапки документа	Таблица с реквизитами многострочной части документа	Таблица с реквизитами серийных номеров	Таблица с реквизитами номеров партий
001	Приходная накладная	spr001	spec001	barcodes	barcodes2
002	Расходная накладная	spr002	spec002	barcodes	barcodes2
003	Списание / оприходование ТМЦ	spr003	spec003	barcodes	barcodes2
004	Товарный чек	spr004			
005	Переоценка ТМЦ	spr005	spec005		
006	Возврат от покупателей	spr006	spec006	barcodes	barcodes2
007	Инвентаризация ТМЦ	spr007	spec007	barcodes	barcodes2
008	Документ кассовой смены	spr008	spec008	barcodes	barcodes2
009	Выпуск и комплектация	spr009	spec009	barcodes	barcodes2
010	Счет (исходящий)	spr010	spec010		
011	Коммерческое предложение	spr011	spec011		
012	Выписка из банка	spr012	spec012		
013	Приход наличных денег	spr013			
014	Аванс полученный	spr014			
015	Платежное поручение	spr015			
016	Расходный кассовый ордер	spr016			
017	Расход наличных денег	spr017			
018	Резервирование ТМЦ	spr018	spec018		
019	Внутреннее перемещение	spr019	spec019	barcodes	barcodes2
020	Заказ от покупателя	spr020	spec020		
021	Пересортица ТМЦ	spr021	spec021	barcodes	barcodes2
022	Счет-фактура (исходящий)	spr022	spec022		
023	Приходный кассовый ордер	spr023			
027	Журнал проводок	spr027	spec027		
028	Счет-фактура (входящий)	spr028			
029	Внешнее движение средств	spr029	spec029		
030	Формирование стоимости объекта ОС	spr030	spec030	barcodes	barcodes2
031	Капитализация объектов ОС	spr031	spec031		
032	Счет (входящий)	spr032	spec032		
033	Заказ поставщику	spr033	spec033		
034	Заказ на поставку	spr034	spec034		
036	Поступление ОС	spr036	spec036		
037	Амортизация ОС	spr037	spec037		
038	Движение ОС	spr038	spec038		
039	Ценная бумага (входящая)	spr039	spec039		
040	Ценная бумага (исходящая)	spr040	spec040		
041	Формирование книги покупок	spr041	spec041		
042	Перемещение стоимости ОС	spr042	spec042	barcodes	
043	Инвентаризация ОС	spr043	spec043		
044	Авансовый отчет	spr044	spec044		
049	Акт сдачи/приемки услуг (входящий)	spr049	spec049		



050	Акт сдачи/приемки услуг (исходящий)	spr050	spec050		
052	Заявка на оплату	spr052	spec052		
053	Инкассация	spr053	spec053		
054	План - меню	spr054	spec054		
055	Возврат поставщику	spr055	spec055	barcodes	barcodes2
056	Калькуляция	spr056	spec056		
057	Демонтаж ОС	spr057	spec057	barcodes	barcodes2

Как видно из таблицы, имена таблиц базы данных соответствуют кодам типов документов, которые хранятся в этих таблицах.

С полным перечнем базовых реквизитов документов можно ознакомиться в приложении №...

### **Дополнительные таблицы документов.**

Как было сказано выше, для каждого типа документов в системе Айтида предусмотрены свои таблицы для хранения базовых реквизитов документов. Кроме того, предусмотрены таблицы, в которых хранятся однородные для многих типов документов реквизиты, такие как, например, серийные номера или номера партий. Ниже приведен список дополнительных таблиц, содержащих важную информацию о документах.

<b>Имя таблицы</b>	<b>Назначение</b>
spr000	Журнал всех документов.
barcodes	Серийные номера товаров .
barcodes2	Номера партий товаров.
docsanalit_ex	Дополнительные реквизиты документов, определенные в точке зрения.
spec000_img	Дополнительные файлы, прикрепленные к документу.
statuslist	История изменения статусов документов.
accountlist	Список моделей учета, на которые должен влиять документ.
crosslink	Таблица связей между документами.

### **Журнал всех документов.**

Для доступа к наиболее часто встречающимся реквизитам документов и для упрощения процедур формирования общего списка всех имеющихся в системе документов предусмотрена таблица spr000. Данная таблица формируется в системе автоматически. Также, автоматически происходит ее обновление в момент внесения изменений в основную таблицу документа. Изменения производятся триггерами, поэтому не важно, каким образом была изменена информация в основной таблице, все изменения будут отражены в spr000.

Также, в таблице spr000 хранятся системные реквизиты документов, такие как: признак печати документа, признак выгрузки документа в ККМ, признак отправки документа в другой филиал, признак помеченного на удаления документа, признак запрета повторяющихся строк, номер автоматически сгенерированной партии, код филиала создавшего документ, идентификатор документа в филиале, создавшем документ.

Таблица spr000 играет важную роль при обмене данными между филиалами. По полям «код филиала создавшего документ» и «идентификатор документа в филиале, создавшем документ», происходит идентификация документов при приеме данных из другого филиала. Эти поля заполняются автоматически при создании нового документа и, впоследствии, не изменяются.

Записи в таблице spr000 создаются в момент добавления новых записей в таблицы шапок документов. В дальнейшем, происходит автоматическое обновление в соответствии с вносимыми

в исходные документы изменениями. При удалении записей из исходных таблиц, соответствующие записи из таблицы spr000 также удаляются.

Имя поля и тип данных	Назначение	Наличие поля в основной таблице
code char(3)	Код типа документа, см. таблицу docstab. По этому полю определяется, в какой таблице находятся исходные данные документа. Заполняется один раз при добавлении записи и впоследствии не изменяется.	
ic int	Идентификатор документа в локальной базе. Соответствует полю identity_column соответствующей таблицы шапки документа. Заполняется один раз при добавлении записи и впоследствии не изменяется.	✓
ndok char(20)	Номер исходного документа. Изменяется автоматически при изменении исходной записи документа.	✓
date datetime	Дата исходного документа. Изменяется автоматически при изменении исходной записи документа.	✓
firm char(7)	Код организации, от имени которой выписан документ. Изменяется автоматически при изменении исходной записи документа.	✓
rs char(20)	Расчетный счет организации. Изменяется автоматически при изменении исходной записи документа.	✓
client char(7)	Код контрагента. Изменяется автоматически при изменении исходной записи документа.	✓
contract char(10)	Код договора. Изменяется автоматически при изменении исходной записи документа.	✓
cur char(3)	Код валюты документа. Изменяется автоматически при изменении исходной записи документа.	✓
cur_rate float	Курс конвертации валюты документа в основную валюту учета документа. Изменяется автоматически при изменении исходной записи документа.	✓
f_manualrate bit	Признак ручного ввода курса конвертации валюты документа. Если данный признак не установлен, то при записи документа, система автоматически обновит курс конвертации на значение из справочника курсов валют.	
note varchar(250)	Комментарий к документу. Изменяется автоматически при изменении исходной записи документа.	✓
author varchar(250)	Код пользователя, являющегося автором документа. Данное поле заполняется автоматически в момент создания нового документа и больше не изменяется	✓
branch char(3)	Код филиала, владельца документа. В исходном документе, данное поле заполняется из используемых в документе объектов – склада, денежного кармана и т.п. Вручную не устанавливается. Изменяется автоматически при изменении исходной записи документа.	✓
branch_d char(3)	Код филиала, в который необходимо передать документ. В исходном документе, данное поле заполняется из используемых в документе объектов – склада получателя, денежного кармана получателя и т.п. Вручную не устанавливается. Изменяется автоматически при изменении исходной записи документа.	✓
empl char(7)	Код сотрудника. Изменяется автоматически при изменении исходной записи документа.	✓
summa float	Сумма документа. Итоговая сумма документа. Изменяется автоматически при изменении исходной записи документа.	✓
sklad char(3)	Код склада или склада отправителя. Изменяется автоматически при изменении исходной записи документа.	✓
sklad_d char(3)	Код склада получателя. Изменяется автоматически при изменении исходной записи документа.	✓
manager char(7)	Код ответственного менеджера из справочника сотрудников. Изменяется автоматически при изменении исходной записи документа.	✓
maindb char(3)	Код филиала, в котором был изначально создан документ. Заполняется автоматически в момент создания документа или в момент приема данных из другого филиала.	
mainic int	Идентификатор документа в филиале, в котором документ был изначально создан. Заполняется автоматически в момент создания	

	документа или в момент приема данных из другого филиала.	
f_print bit	Признак того, что документ был распечатан. Системой не используется. Может быть установлен в момент печати формы документа и отображен в соответствующем журнале документов.	
f_kkm bit	Признак того, что документ был выгружен в оборудование. Устанавливается формой обмена данными с оборудованием после выгрузки данных в оборудование.	
f_sent bit	Признак того, что документ был передан в другой филиал.	
f_deleted bit	Признак того, что документ был помечен на удаление.	
f_singlennline bit	Признак того, что в документе запрещены строки с одинаковыми товарами/услугами.	
partnumber varchar(50)	Номер партии указанный для всего документа или сгенерированный автоматически.	

Примечания.

1. При изменении полей client, contract, summa, cur, cur\_rate триггер на изменение записей вызывает функцию dbo.fn\_checklinks, которая проверяет наличие сопоставлений у изменяемого документа. Если сопоставления с другими документами есть, то изменение указанных полей запрещается и транзакция отклоняется. В системе выдается соответствующее сообщение.
2. В момент удаления документа, триггер на удаление записей проверяет признак f\_sent (документ передан в другой филиал). Если данный признак установлен, то удаление запрещается, транзакция на удаление отклоняется и в системе выдается соответствующее сообщение. Если все проверки прошли успешно, то автоматически очищаются следующие таблицы:
  - accountlist
  - spec000\_img
  - statuslist
  - crosslink
  - docsanalit\_ex
  - spectree
  - Если документ был создан из кассовых транзакций, то в таблице транзакций (frontol\_\_tranzt) освобождаются (поля code и ic устанавливаются в "пустые" значения) использованные транзакции и по ним (транзакциям) могут быть созданы новые документы.

### Серийные номера товаров (barcodes).

Для организации ввода и хранения первичной информации для учета серийных номеров товаров предусмотрена таблица barcodes. Эта таблица содержит исходную информацию для всех типов документов, в которых возможно наличие серийных номеров. Для ввода пользователем доступны два поля Серийный номер (barcode) и количество (kolp). Остальные поля заполняются системой автоматически, на основании того, для какой строки и в каком документе введены серийные номера.

Имя поля и тип данных	Назначение
code char(3)	Код типа документа, см. таблицу docstab. По этому полю определяется, тип родительского документа.
ic int	Идентификатор документа. Идентификатор (identity_column) родительского документа.
spec_ic int	Идентификатор строки документа, для которой определены серийные номера. Соответствует полю identity_column таблицы многострочной части родительского документа. Например, для документа Приходная накладная многострочная часть хранится в таблице spec001.
nn char(10)	Внутренний код товара, для которого указаны серийные номера. Заполняется автоматически в момент записи документа на основании кода товара из многострочной части документа. Используется при проведении документа в регистре учета серийных номеров.
barcode varchar(50)	Введенный серийный номер. Сохраняется в том виде, как был введен пользователем.
kolp float	Количество серийных номеров. Сумма количеств всех серийных номеров для одной строки документа должна соответствовать количеству товара, указанному в этой строке.

## Номера партий товаров (barcodes2).

Для организации ввода и хранения первичной информации для учета номеров партий товаров предусмотрена таблица `barcodes2`. Эта таблица содержит исходную информацию для всех типов документов, в которых возможно наличие номеров партий.

Система Айтида предусматривает два варианта работы с номерами партий товаров.

1. Номера партий могут быть указаны пользователем "вручную", или могут быть сгенерированы системой автоматически. В случае ручного ввода, пользователь самостоятельно заполняет поля Номер партии (`barcode`) и Количество (`kolp`). В случае прихода, указанные номера партий будут оприходованы. В случае расхода – система спишет товар с указанных пользователем партий, в случае их наличия. При этом принцип FIFO может не соблюдаться.
2. Система может автоматически сгенерировать номера партий при приходе товаров. И автоматически распределить товар по имеющимся в наличии номерам партий при расходе товаров. При этом, в зависимости от настроек, система будет либо строго следовать принципу FIFO, либо списывать товар по упрощенной схеме FIFO, при которой корректировки документов "задним" числом могут приводить к нарушению данной схемы. Все автоматически сгенерированные номера партий будут сохранены в таблице `barcodes2`.

Имя поля и тип данных	Назначение
<code>code char(3)</code>	Код типа документа, см. таблицу <code>docstab</code> . По этому полю определяется, тип родительского документа.
<code>ic int</code>	Идентификатор документа. Идентификатор ( <code>identity_column</code> ) родительского документа.
<code>spec_ic int</code>	Идентификатор строки документа, для которой определены номера партий. Соответствует полю <code>identity_column</code> таблицы многострочной части родительского документа. Например, для документа Приходная накладная многострочная часть хранится в таблице <code>spec001</code> .
<code>nn char(10)</code>	Внутренний код товара, для которого указаны серийные номера. Заполняется автоматически в момент записи документа на основании кода товара из многострочной части документа. Используется при проведении документа в регистре учета серийных номеров.
<code>barcode varchar(50)</code>	Введенный серийный номер. Сохраняется в том виде, как был введен пользователем.
<code>kolp float</code>	Количество серийных номеров. Сумма количеств всех серийных номеров для одной строки документа должна соответствовать количеству товара, указанному в этой строке.
<code>fauto bit</code>	Признак того, что номер партии был сгенерирован системой автоматически во время принятия документа к учету.
<code>reg006_ic int</code>	Ссылка на идентификатор ( <code>identity_column</code> ) записи в регистре учета номеров партий ( <code>reg006</code> ), которая создала указанную запись с "автоматическим" номером партии.

### Примечание.

1. В момент принятия приходного документа к учету (проведении), при необходимости ведения учета партий товаров, система Айтида дополняет список указанных пользователем номеров партий товара необходимым количеством автоматически сгенерированных номеров, таким образом, чтобы суммарное количество номеров партий строго соответствовало количеству товара в документе. Номера партий генерируются либо из поля `partnumber` таблицы `spr000`, если были указаны, либо исходя из типа и идентификатора принимаемого к учету документа.
2. В момент принятия расходного документа к учету (проведении), при отсутствии указаний пользователя на конкретные номера партий, подлежащих к списанию, система будет производить автоматическое распределение, имеющихся в наличии номеров партий таким образом, чтобы более ранние оприходованные партии были списаны как можно раньше (схема FIFO). В процессе распределения все обработанные номера партий будут добавлены в таблицу `barcodes2` и будут доступны для просмотра в документе. В случае нехватки необходимого количества номеров партий проведение документа будет отменено и выдано соответствующее сообщение. Все произведенные изменения в базе данных, так же будут отменены.

## Дополнительные реквизиты документов, определенные в точке зрения (docsanalit\_ex).

Базовые реквизиты документов в большинстве случаев позволяют адекватно и полно передать системе всю необходимую для ведения учета информацию. Но бывают случаи, когда особенности ведения бизнеса в Организации требуют для ведения учета анализа дополнительных, не предусмотренных в базовой конфигурации системы, параметров. Для удовлетворения подобных требований система Айтида предоставляет возможность добавлять в документы, как в шапку документа, так и в многострочную часть, необходимые реквизиты. Добавление новых реквизитов производится без изменения структуры таблиц базы данных. Значения всех новых реквизитов записываются в специальную таблицу – docsanalit\_ex.

Дополнительные реквизиты делятся на две группы:

1. Реквизиты, доступные всем пользователям системы и влияющие на все модели учетов. Такие реквизиты добавляются в точке зрения на закладке Расширенные реквизиты/свойства/ команды. Описание дополнительных реквизитов в точке зрения производится с использованием Справочника параметров и полей ввода.
2. Реквизиты, которые необходимы только для конкретных моделей учета, и которые не должны быть доступны пользователям других моделей учета. Такие реквизиты добавляются в справочнике бизнес-функций на закладке Реквизиты. Описание дополнительных реквизитов в бизнес – функции производится с использованием Справочника понятий.

Имя поля и тип данных	Назначение
code char(3)	Код типа документа, см. таблицу docstab. По этому полю определяется, тип документа, к которому относится реквизит.
ic int	Идентификатор документа. Идентификатор (identity_column) документа, к которому относится реквизит.
spec_ic int	Идентификатор строки документа, для которой указан реквизит. Соответствует полю identity_column таблицы многострочной части документа
account_char(3)	Код модели учета, для которой указан реквизит. Для реквизитов, указанных в точке зрения это поле содержит строку из трех символов - ("---").
t_concept varchar(100)	Имя понятия, описывающего дополнительный реквизит. Данное поле используется, если дополнительный реквизит был описан в бизнес – функции документа.
parameter varchar(250)	Имя параметра, описывающего дополнительный реквизит. Данное поле используется, если дополнительный реквизит был описан в точке зрения.
analit char(13)	Значение реквизита, в случае, если выбор производился из справочника. Первые три символа содержат код из справочника аналитик (spranalit), последующие символы содержат код выбранной карточки используемого справочника.
txt varchar(250)	Значение реквизита, в случае ввода текстового поля.
flt float	Значение реквизита, в случае ввода числового или логического значения.
dt datetime	Значение реквизита, в случае ввода значения типа дата и время.
f_nf bit	Признак, если 1, то при записи документа значение реквизита обязательно должно быть указано. Используется при описании дополнительного реквизита в бизнес-функции.
label_varchar(250)	Метка, указываемая у поля ввода значения реквизита. Используется при описании дополнительного реквизита в бизнес-функции.
fnum int	Номер дополнительного реквизита в списке дополнительных реквизитов., для организации заданного порядка ввода значений. Используется при описании дополнительного реквизита в бизнес-функции.
value_txt text	Текстовое представление значения реквизита, которое можно передавать вычислителю. Например, для дополнительного реквизита с именем ДополнительныйРеквизит и значением в поле analit = " cln0010076", в поле value_txt будет указано: <b>ДополнительныйРеквизит_список="0010076";ДополнительныйРеквизит_папки="0";ДополнительныйРеквизит="0010076";ДополнительныйРеквизит_имя="ООО Чемоданчик";</b> Эту строку можно передать вычислителю, для определения соответствующих переменных с указанными значениями.

### Дополнительные файлы, прикрепленные к документу (spec000\_img).

Система Айтида предусматривает возможность сохранения в базе данных различных внешних, по отношению к системе, файлов связанных с документом. Примерами таких файлов могут служить оригиналы накладных или актов в отсканированном виде, или исходные файлы с данными, из которых были сформированы электронные документы. В пользовательском интерфейсе предусмотрены механизмы загрузки таких файлов для всех типов документов. В базе данных эти файлы хранятся в специальной таблице spec000\_img.

Имя поля и тип данных	Назначение
code char(3)	Код типа документа, см. таблицу docstab. По этому полю определяется, тип документа, к которому относится прикрепленный файл.
ic int	Идентификатор документа. Идентификатор (identity_column) документа, к которому относится прикрепленный файл..
pnum int	Номер файла по порядку, для правильного отображения в списке файлов
doc_image image	Содержимое файла в двоичном виде. В это поле исходный файл загружается целиком
filename varchar(250)	Имя исходного файла. При просмотре файла из интерфейса системы, во временном каталоге создается файл с указанным именем и функции по его открытию передаются операционной системе, которая по расширению имени файла определяет, какое приложение должно открыть указанный файл.
f_main bit	Признак, который может установить пользователь.

**Примечание.** Функционал систем Айтида позволяет загружать в качестве внешних файлов специально подготовленные шаблоны документов, содержащие управляющие тэги, которые могут быть преобразованы и сохранены как готовые внешние документы. Такие шаблоны должны быть подготовлены в формате MS Word XML.

### История изменения статусов документов (statuslist).

Система Айтида предусматривает механизмы для работы со статусами документов. Под статусом понимается состояния документа во внутреннем документообороте Организации с правилами перехода документа из одного состояния в другое. Так же, система может контролировать различные операции, разрешенные или запрещенные для документов в определенном статусе. Статус документа определяется для конкретной модели учета. Правила работы со статусами документов задаются в **Справочнике статусов документов**. Изменение статуса документа производится из интерфейса системы в форме журнала соответствующего типа документов. История изменения статусов документов хранится в таблице statuslist.

Имя поля и тип данных	Назначение
code char(3)	Код типа документа, см. таблицу docstab. По этому полю определяется, тип документа, к которому относится статус.
ic int	Идентификатор документа. Идентификатор (identity_column) документа, к которому относится статус.
account_char(3)	Учет, для которого был изменен статус документа.
status char(3)	Статус документа, в котором находится документ.
_status char(3)	Статус документа, в котором находился документ до перехода в текущий статус.
fback bit	Признак, если <b>ИСТИНА</b> , возврата в прежний статус.
back_ic	Идентификатор отменной строки истории изменения статуса при возврате.
canceled	Признак, если <b>ИСТИНА</b> , того что из данного статуса был произведен возврат.
posorder int	Порядковый номер. Документ может принимать один и тот же статус много раз. Данное поле используется системой для определения порядка при возврате документа в предыдущий статус.
statustime datetime	Дата и время перехода документа в текущий статус.
author varchar(128)	Пользователь, присвоивший статус документу.

**Примечание.** В системе Айтида немного различаются процедуры перевода документа в следующий статус и возврат в предыдущий статус. Различие заключается в том, что при возврате в предыдущий статус в историю изменения статусов записывается признак возврата в прежний статус. Предусмотрена возможность запрета возврата в предыдущий статус.

### Список моделей учета, на которые должен влиять документ (accountlist).

Система Айтида предоставляет возможность одновременного ведения учета в нескольких моделях на одних и тех же исходных документах. Данная возможность реализована привязкой каждого конкретного документа к требуемым моделям учета. В каждой модели документ может иметь различные правила отражения в учете, это достигается указанием для каждой модели учета собственной бизнес – функции документа, может иметь свой статус, свои сопоставления с другими документами. Все эти возможности позволяют максимально независимо вести учет в различных моделях на единых исходных документах.

Имя поля и тип данных	Назначение
code char(3)	Код типа документа, см. таблицу docstab.
ic int	Идентификатор документа. Идентификатор (identity_column) документа, к которому относится модель учета.
account_ char(3)	Код модели учета, в которой участвует указанный документ.
f_main bit	Признак, того что документ был последний раз сохранен с этим учетом.
bfn char(10)	Код бизнес – функции, которую выполняет документ в модели учета.
transacted bit	Признак, если 1, то документ принят к указанному учету (проведен).
summa_p float	Общая сумма сопоставлений документа в указанной модели учета.
status char(3)	Текущий статус документа в модели учета.

**Примечание.** Управление содержимым данной таблицы производится либо на основании доступных учетов в точке зрения пользователя, создающего документ, либо вручную через интерфейс управления учетами в форме документа. Содержимое данной таблицы имеет определяющее значение на то, как введенный или загруженный документ будут влиять на систему. Отсутствие записей делает невозможным отражение документа в модели учета.

### Таблица связей между документами (crosslink ).

В системе Айтида предусмотрен механизм, позволяющий создавать логические связи между документами. Применение этого механизма очень удобно при выписке документов, относящихся к одной сделке. Примером может служить типичный пакет документов, описывающий сделку с покупателем. Первым документом является коммерческое предложение, выписанное менеджером, в котором описывает условия поставки товара и цены. В случае согласия покупателя на основании коммерческого предложения выписывается счет на оплату товара. Бухгалтер или кассир выписывает документ об оплате счета (Приход безналичных денег или приходный кассовый ордер). После оплаты счета, производится отгрузка товара расходной накладной, которая оформляется на складе. Вместе с расходной накладной выписывается счет-фактура. В системе вся эта последовательность действий может быть записана и сохранена. Для этого предусмотрено три типа связей между документами: связь *«На основании»*, связь *«Сопоставление - оплата»* и связь *«Сопоставление – отгрузка/оприходование»*.

#### *Связь «на основании»*

Связь «На основании» не оказывает влияния на проведение документов. Система просто запоминает, что данные документы связаны между собой. Связь данного типа может быть создана между двумя любыми документами. При создании нового документа на основании существующего можно выбрать любой тип документа, указать, нужно ли переносить данные из родительского документа.

#### *Связь «сопоставление – оплата»*

Данный тип связи используется, когда необходимо сопоставить «входящие» со стороны контрагента и «исходящие». Под «входящими» документами понимаются документы поставки товарно-материальных ценностей или оплаты со стороны контрагента. Под «исходящими»

документами понимаются документы отгрузки товарно-материальных ценностей или оплаты контрагентам. Данный тип связи влияет на учет взаиморасчетов с контрагентами. Система поддерживает установку связи между документами в разных валютах с корректным расчетом суммовых и курсовых разниц. На основании данных связей рассчитываются остатки по взаиморасчетам с контрагентами и, в случае необходимости, переоценка остатков в валюте задолженности. В разных моделях учета один документ может быть сопоставлен с разными документами.

*Связь «сопоставление – отгрузка/поставка»*

Данный тип связи похож на связь «сопоставление – оплата», но доступен только для документов Счет (входящий) и Счет (исходящий), и может применяться только в отношении приходной накладной (для входящего счета) и расходной накладной (для исходящего счета). Данный тип связи не поддерживает расчетов в разных валютах, не рассчитывает суммовые и курсовые разницы.

Имя поля и тип данных	Назначение
code char(3)	Код типа исходного документа, см. таблицу docstab.
ic int	Идентификатор исходного документа. Идентификатор (identity_column) документа, к которому относится связь.
ndok char(20)	Номер исходного документа.
data datetime	Дата исходного документа.
spec_ic int	Идентификатор строки многострочной части исходного документа, в случае сопоставления не со всем документом, а только с отдельной строкой. Например, документ Банковская выписка, позволяет сопоставлять отдельные строки с другими документами.
cr_code char(3)	Код типа связанного документа, см. таблицу docstab.
cr_ic int	Идентификатор связанного документа. Идентификатор (identity_column) документа, к которому относится связь.
cr_ndok char(20)	Номер связанного документа.
cr_data datetime	Дата связанного документа.
cr_spec_ic int	Идентификатор строки многострочной части связанного документа, в случае сопоставления не со всем документом, а только с отдельной строкой. Например, документ Банковская выписка, позволяет сопоставлять отдельные строки с другими документами.
date_link datetime	Дата и время связи
cur char(3)	Валюта сопоставляемого документа.
cur_rate float	Курс пересчета валюты сопоставляемого документа.
summa float	Сумма сопоставления.
gain float	Рассчитанная суммовая разница.
cr_type char(1)	Тип связи 0- на основании, 1- сопоставление – оплата, 2- сопоставление – отгрузка/поставка.
account_char(3)	Модель учета, для которой создана связь.
fstorno bit	Признак, если 1, то данная связь является сторнирующей по отношению к исходной.
stornoic int	Идентификатор сторнированной связи.

**Примечание.**

1. Связь «на основании» задействует только поля, идентифицирующие связываемые документы. В поле account\_ в этом случае записывается значение ---. Связь «на основании» не может быть сторнирована, т.к. она не влияет ни на одну модель учета.
2. Связи «сопоставление» влияют на указанную модель учета. По умолчанию, при сопоставлении документов датой сопоставления, и, соответственно, датой которой связь будет влиять на модель учета, считается наибольшая из дат двух сопоставляемых документов. В случае если необходимо указать специальную дату, то она будет записана в поле date\_link. Всё влияние связи на модель учета будет произведено именно этой датой.
3. Связь с типом «сопоставление» самостоятельно влияет на регистр взаиморасчетов с контрагентами. Отмена проведения документа не удаляет это влияние. Только при отмене сопоставления данное влияние будет отменено. В случае необходимости отменить связь «сопоставление» в закрытом для корректировок периоде, необходимо пользоваться механизмом сторнирования сопоставлений. В этом случае, информация об исходном сопоставлении будет указана в поле stornoic.



4. При создании связи «на основании» в таблицу crosslink добавляется одна запись указывающая на оба связываемых документа. При создании связи «сопоставления» добавляется две записи – по одной для каждого сопоставляемого документа. Тем не менее, при удалении одно из них, вторая будет удалена автоматически.

## Пример заполнения таблиц документа.

Рассмотрим на примере нескольких документов то, как и чем заполняются вышеописанные таблицы данными, при выполнении различных операций с документами.

1. Создадим документ Приходная накладная, как указано на картинке:

Приходная накладная - корректировка № А-0 от 07.09.2011 - УПР учет

Основные реквизиты: ООО "ОАЗИС" Номер: А-0 Дата: 07.09.2011 11:30:03

Контрагент: ОАО Хлебный Дом Склад: Склад Договор: не выбран

Сумма: 1760.00 RUB ( 1760.00 у.е.) НДС: 160.00 ( 160.00 у.е.)

№ п/п	Т	Код	Наименование	Серия	Номер	ДР	Кол-во	Цена	Сумма	НДС	Единица	Налог	Проц
1		426	Батон "Городской" в\с 400 г				15.000	13.00	195.00	17.72	шт	НДС 10%	Проц
2		439	Пирожок с капустой 80 г				25.000	11.00	275.00	25.00	шт	НДС 10%	Проц
3		430	Хлеб "Дарницкий" нар. 630 г				50.000	16.00	800.00	72.72	шт	НДС 10%	Проц
4		431	Хлеб "Докторский" с отрубями 300 г				20.000	17.00	340.00	30.90	шт	НДС 10%	Проц
5		426	Батон "Городской" в\с 400 г				10.000	15.00	150.00	13.63	шт	НДС 10%	Проц

Дополнительные поля документа

Записать Закрыть

Приходная накладная - корректировка № А-0 от 07.09.2011 - УПР учет

Основные реквизиты: Срок оплаты: 20.02.2008 ОС Вид: Закупка Тара в залог: Нет Скидка: 0.0000

Ответственный: не выбран

Период: Оплаченная сумма: 0.00 Автор: Администратор

Распределить стоимость услуг по ТМЦ  Не учитывать стоимость услуг в сумме документа

№ п/п	Т	Код	Наименование	Серия	Номер	ДР	Кол-во	Цена	Сумма	НДС	Единица	Налог	Проц
1		426	Батон "Городской" в\с 400 г				15.000	13.00	195.00	17.72	шт	НДС 10%	Проц
2		439	Пирожок с капустой 80 г				25.000	11.00	275.00	25.00	шт	НДС 10%	Проц
3		430	Хлеб "Дарницкий" нар. 630 г				50.000	16.00	800.00	72.72	шт	НДС 10%	Проц
4		431	Хлеб "Докторский" с отрубями 300 г				20.000	17.00	340.00	30.90	шт	НДС 10%	Проц
5		426	Батон "Городской" в\с 400 г				10.000	15.00	150.00	13.63	шт	НДС 10%	Проц

Дополнительные поля документа

Записать Закрыть

Прочая информация о поставщике

№ счета-фактуры: 000000001  
Дата сч-фактуры: 07.09.2011  
№ машины: ЯЯ 000 ЯЯ 00  
№ накладной: 000000001  
Дата накладной: 07.09.2011  
№ уведомления:   
Дата уведомления: 01.01.1900

Итого по документу

Параметр	Значение
Сумма товаров	1760.00
Товары с НДС 10	1760.00
НДС 10%	160.00

Комментарий

Пример документа Приходная накладная

Записать Отмена Сохранить Отмена

Ввод номеров

Батон "Городской" в\с 400 г

Серийные номера	Количество
4606471115715	15.000

Итого количество: 15.000

Записать Отмена

Ввод номеров

Батон "Городской" в\с 400 г

Номер партии	Количество	Дата партии
ПАРТИЯ000001	15.000	07.09.2011 00:00

Итого количество: 15.000

Записать Отмена

Работа с приложениями к документу

Приложения к документу

Имя файла документа	Размер документа
Скан001.jpg	97003

Ввод расширенных реквизитов

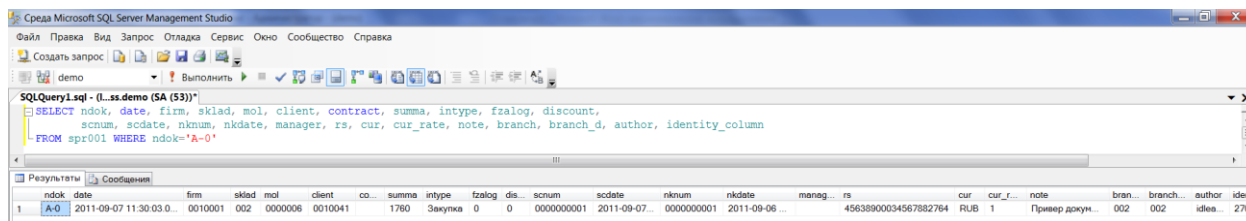
Внешний вид: Удовлетворительный

Записать Отмена

Записать Отмена

После записи этого документа в базу данных будет добавлена следующая информация. Для просмотра данных необходимо запустить среду MS SQL Server, подключиться к серверу и в окне запроса ввести и выполнить запрос:

```
SELECT ndok, date, firm, sklad, mol, client, contract, summa, intype,
       fzalog, discount, scnum, scdate, nknum, nkdate, manager, rs,
       cur, cur_rate, note, branch, branch_d, author, identity_column
FROM spr001 WHERE ndok='A-0'
```



В результате выполнения запроса получим строку шапки нашего документа. Рассмотрим подробнее полученный результат.

Имя поля	Значение	Комментарий
ndok	A-0	Номер документа. При создании нового документа пользователь может самостоятельно указать номер документа или оставить это поле пустым. В этом случае система автоматически присвоит следующий номер документу, удовлетворяющий настройкам: периоду нумерации, указанному префиксу и заданной длине.
date	2011-09-07 11:30:03.000	Дата документа. Основная дата документа. При создании документа используется текущая дата и время. Может быть установлена пользователем. Данной датой документ будет отражен в учете.
firm	0010001	Код Организации, от имени которой выписан документ.
sklad	002	Код склада, на который будет оприходован товар. В данном примере - это Склад, находящийся в филиале № 2 (см. поле branch)/
mol	0000006	Код материально ответственного лица, который приходит товар.
client	0010041	Код контрагента, от которого получен товар.
contract		Код договора контрагента, по которому получен товар. В данном случае договор не указан. Поле пустое.
summa	1760	Общая сумма принятого товара по документу. Данная сумма всегда с учетом всех налогов.
intype	Закупка	Вид документа (закладка <i>Дополнительные реквизиты/Вид</i> )
fzalog	0	Признак Тара в залог (закладка <i>Дополнительные реквизиты/Тара в залог</i> ). 0 соответствует Нет, 1 – Да.
discount	0	Процент примененной скидки. В данном случае скидка не использовалась (закладка <i>Дополнительные реквизиты/Скидка</i> )
scnum	0000000001	Номер счета-фактуры поставщика.
scdate	2011-09-07 00:00:00.000	Дата счета-фактуры поставщика. Дата указывается без времени.
nknum	0000000001	Номер накладной поставщика.
nkdate	2011-09-06 00:00:00.000	Дата накладной поставщика. Дата указывается без времени.
manager		Код менеджера, ответственного за поставку. Код из справочника сотрудников. В данном примере менеджер не указан.
rs	45638900034567882764	Расчетный счет «нашей» организации. Может быть использован для оформления печатных форм документов.
cur	RUB	Валюта документа. В данном примере - рубль.
cur_rate	1	Курс валюты документа к валюте учета.
note	Пример документа Приходная накладная	Комментарий указанный в окне комментария (кнопка <i>Комментарий</i> в панели инструментов документа).
branch	002	Код филиала, для которого выписан документ. В документе Приходная накладная данный код берется из карточки склада, выбранного в поле Склад.
branch_d	002	Для документа Приходная накладная филиал получатель (branch_d) всегда совпадает с филиалом документа (branch). Для документов, которые перемещают ТМЦ (внутреннее перемещение, выпуск готовой продукции, пересортица) или денежные средства (приходный

		кассовый ордер, расходный кассовый ордер, приход наличных денег, расход наличных денег, товарный чек) между филиалами эти поля могут иметь разные значения.
author	idleadadmin	Автор документа. Код из справочника сотрудников.
identity_column	270	Идентификатор документа. Данное значение будет использовано в последующих запросах.

Теперь выполним запрос для просмотра многострочной части документа и рассмотрим подробнее первую строку результата. Обратите внимание на условие отбора записей. Оно содержит идентификатор документа (см. предыдущую таблицу).

```
SELECT ic, nn, s_code, ed, kolp, inside, packs, realkolp, koef_e, koef_c,
      ftara, cena, summa, sumnds, sklad, project, kodn, nnname, declar,
      date_ex, linenum, identity_column
FROM spec001 WHERE ic= 270
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query window displays the SQL query: `SELECT ic, nn, s_code, ed, kolp, inside, packs, realkolp, koef_e, koef_c, ftara, cena, summa, sumnds, sklad, project, kodn, nnname, declar, date_ex, linenum, identity_column FROM spec001 WHERE ic= 270`. The results pane shows a table with 15 columns and 5 rows of data.

ic	nn	s_code	ed	kolp	inside	packs	realkolp	koef_e	koef_c	ftara	cena	summa	sumnds	sklad	project	kodn	nnname	declar	date_ex	linenum	identity_column
270	000000406		шт	15	1	15	0	1	1	0	13	195	17,72727273	002	0000000001	01	Батон "Городской" в\с 400 г		1900-01-01 00:00:00.000	1	2631
270	000000419		шт	25	1	25	0	1	1	0	11	275	25	002	0000000001	01	Пирожок с капустой 80 г		1900-01-01 00:00:00.000	2	2632
270	000000410		шт	50	1	50	0	1	1	0	16	800	72,72727273	002	0000000001	01	Хлеб "Дарницкий" нар. 630 г		1900-01-01 00:00:00.000	3	2633
270	000000411		шт	20	1	20	0	1	1	0	17	340	30,90909091	002	0000000001	01	Хлеб "Докторский" с отрубями 300 г		1900-01-01 00:00:00.000	4	2634
270	000000406		шт	10	1	10	0	1	1	0	15	150	13,63636364	002	0000000001	01	Батон "Городской" в\с 400 г		1900-01-01 00:00:00.000	5	2635

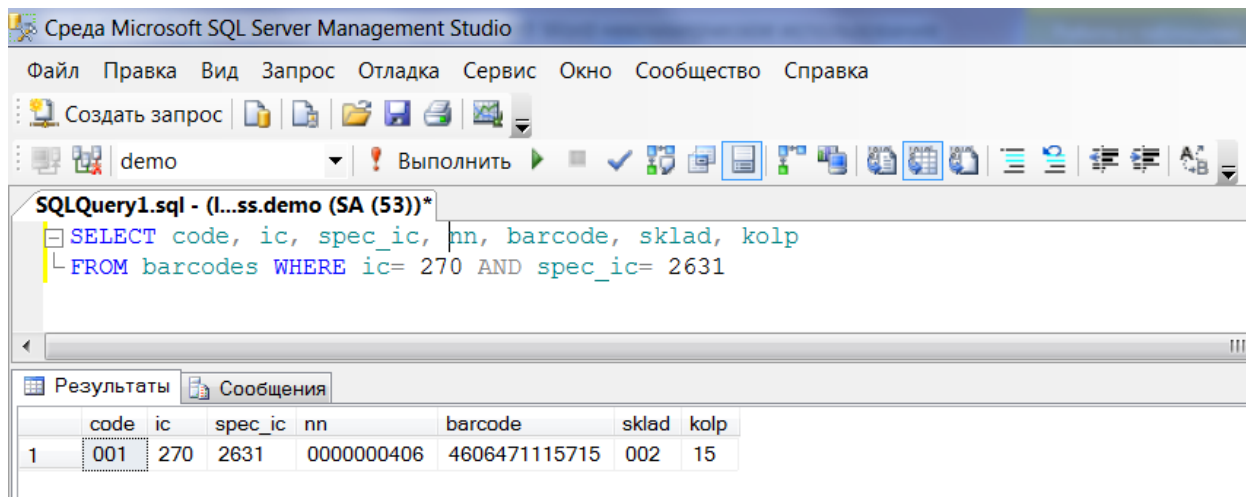
Имя поля	Значение	Комментарий
ic	270	Идентификатор документа, которому принадлежит строка.
nn	000000406	Внутренний код (не путать с кодом в главной базе) товара.
s_code		Внутренний код услуги.
ed	шт	Код единицы измерения.
kolp	15	Количество товара / услуги в указанной единице измерения.
inside	1	Количество единиц товара в одной упаковке.
packs	15	Количество упаковок товара.
realkolp	0	Реальное количество товара, загруженное из ТСД.
koef_e	1	Коэффициент пересчета количества указанной единицы измерения к основной единице измерения товара/услуги.
koef_c	1	Коэффициент пересчета цены указанной единицы измерения к основной единице измерения товара/услуги.
ftara	0	Признак того, что товар являет возвратной тарой. В данном случае не является.
cena	13	Цена товара/услуги, указанная в строке документа. Цена содержит все налоги.
summa	195	Сумма, товара/услуги, указанная в строке документа. Сумма содержит все налоги.
sumnds	17,72727273	Сумма НДС, указанная в строке документа.
sklad	002	Код склада, на который будет оприходован товар. По умолчанию, равен складу, выбранному в шапке документа.
project	0000000001	Код проекта, для которого осуществляется поставка.
kodn	01	Код списка налогов товара/услуги для расчета сумм налогов.
nnname	Батон "Городской" в\с 400 г	Наименование товара/услуги. Сохраняется в документе. При изменении наименования в справочниках остается неизменным.
declar		Номер ГТД.
date_ex	1900-01-01 00:00:00.000	Срок годности партии товара. Дата 1900-01-01 соответствует не указанной «пустой» дате.
linenum	1	Номер строки в документе.
identity_column	2631	Идентификатор строки документа. Данное значение будет использовано в последующих запросах.

Далее рассмотрим остальные таблицы, хранящие информацию о документе.

### Серийные номера.

Теперь выполним запрос для просмотра серийных номеров указанных для строки многострочной части документа и рассмотрим подробнее первую строку результата. Обратите внимание на условие отбора записей. Оно содержит идентификатор документа и идентификатор строки документа (см. предыдущую таблицу).

```
SELECT code, ic, nn, barcode, sklad, kolp
FROM barcodes WHERE ic= 270 AND spec_ic= 2631
```



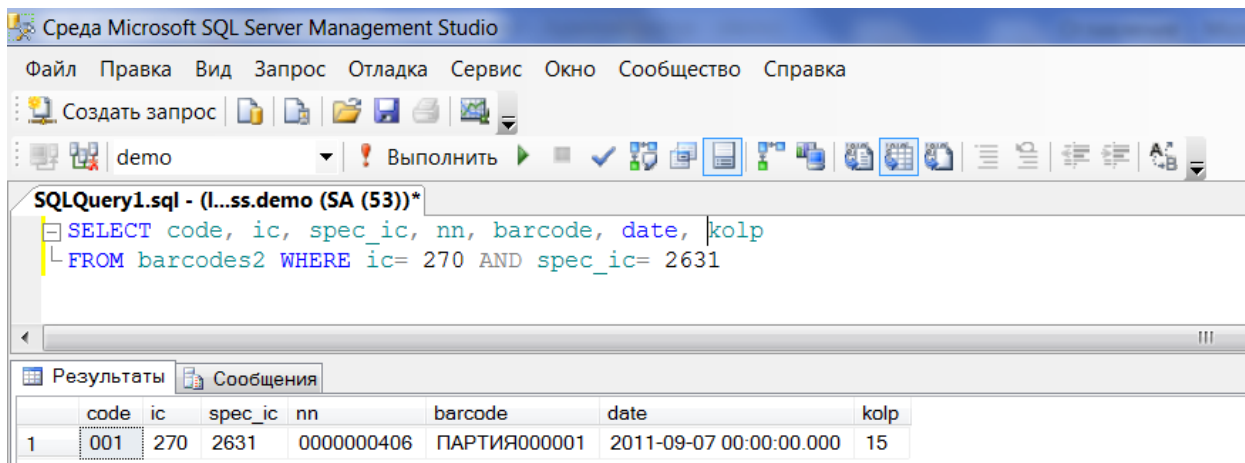
Имя поля	Значение	Комментарий
ic	270	Идентификатор документа, которому принадлежит серийный номер.
spec_ic	2631	Идентификатор строки документа, которой принадлежит серийный номер.
nn	0000000406	Внутренний код товара, которому принадлежит серийный номер.
barcode	4606471115715	Серийные номер.
sklad	002	Склад, из строки документа, на который оприходован данный серийный номер.
kolp	15	Количество оприходованного товара с указанным серийным номером.

**Примечание.** Суммарное количество серийных номеров (сумма по полю kolp), должно совпадать с количеством указанным в строке документа.

### Номера партий.

Теперь выполним запрос для просмотра номеров партий товара указанных для строки многострочной части документа и рассмотрим подробнее первую строку результата. Обратите внимание на условие отбора записей. Оно содержит идентификатор документа и идентификатор строки документа (см. предыдущую таблицу).

```
SELECT code, ic, nn, barcode, sklad, kolp
FROM barcodes2 WHERE ic= 270 AND spec_ic= 2631
```



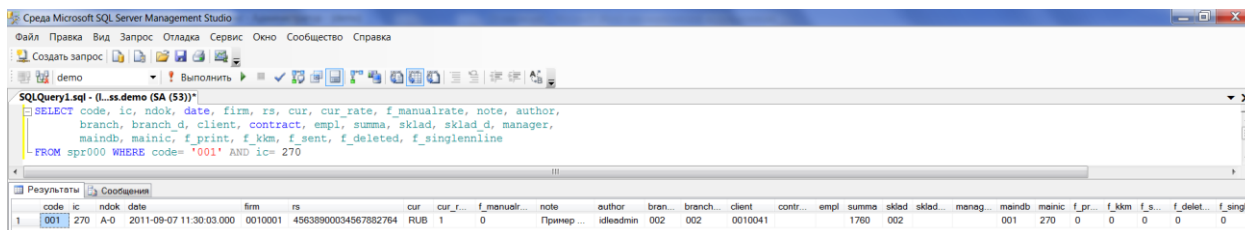
Имя поля	Значение	Комментарий
ic	270	Идентификатор документа, которому принадлежит номер партии.
spec_ic	2631	Идентификатор строки документа, которой принадлежит номер партии.
nn	000000406	Внутренний код товара, которому принадлежит номер партии.
barcode	ПАРТИЯ000001	Номер партии.
date	2011-09-07 00:00:00.000	Дата номера партии. В стандартной конфигурации не используется.
kolp	15	Количество оприходованного товара с указанным номером партии.

Примечание. Если суммарное количество указанных номеров партий (сумма по полю kolp), меньше количества в строке документа, то недостающие номера партий будут сгенерированы автоматически в момент принятия документа к учету. При этом в таблицу barcodes2 они занесены не будут.

### Журнал всех документов.

Теперь выполним запрос для просмотра записей, созданных в журнале всех документов. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code) и идентификатор документа (поле ic). Записи в журнале всех документов создаются автоматически при доавлении записей в таблицы конкретных документов.

```
SELECT code, ic, ndok, date, firm, rs, cur, cur_rate, f_manualrate, note,
author, branch, branch_d, client, contract, empl, summa, sklad, sklad_d,
manager, maindb, mainic, f_print, f_kkm, f_sent, f_deleted,
f_singlennline
FROM spr000 WHERE code= '001' AND ic= 270
```



Имя поля	Значение	Комментарий
code	001	Код типа документа. Устанавливается автоматически при добавлении записи в таблицу spr001 (при добавлении записей в другие таблицы документов, код будет другой).
ic	270	Идентификатор документа.
ndok	A-0	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
date	2011-09-07 11:30:03.000	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
firm	0010001	Поле автоматически копируется из аналогичного поля таблицы шапки

		документа при каждом обновлении записи.
cur	45638900034567882764	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
cur_rate	RUB	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
f_manualrate	1	Признак, если 0, то в момент записи документа система автоатически обновляет курс документа в соответствии со справочником курсов валют.
note	Пример документа Приходная накладная	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
author	idleadmin	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
branch	002	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
branch_d	002	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
client	0010041	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
contract		Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
empl		В Приходной накладной нет такого поля, поэтому для приходной накладной оно всегда пустое.
summa	1760	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
sklad	002	Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
sklad_d		В Приходной накладной нет такого поля, поэтому для приходной накладной оно всегда пустое.
manager		Поле автоматически копируется из аналогичного поля таблицы шапки документа при каждом обновлении записи.
maindb	001	Код филиала, в котором был создан документ. Поле используется при обмене документами между филиалами для идентификации документа.
mainic	270	Идентификатор документа в филиале, в котором был создан документ. Поле используется при обмене документами между филиалами для идентификации документа.
f_print	0	Признак печати документа. Данный документ не был распечатан.
f_kkm	0	Признак выгрузки документа в оборудование. Устанавливается формой обмена с оборудованием в момент выгрузки данных. Данный документ не был выгружен в оборудование.
f_sent	0	Признак отправки документа в другой филиал. Устанавливается процедурой выгрузки данных по шаблону (не путать с выгрузкой в базу копию в центре) в другой филиал.
f_deleted	0	Признак документа, помеченного на удаление. При пометке на удаление документ остается в базе данных, но перестает влиять на все модели учетов. Данный документ не помечен на удаление.
f_singlennline	0	Признак запрета в документе повторяющихся строк – строк с одинаковым товаром/услугой. В данном документе такие строки не запрещены.

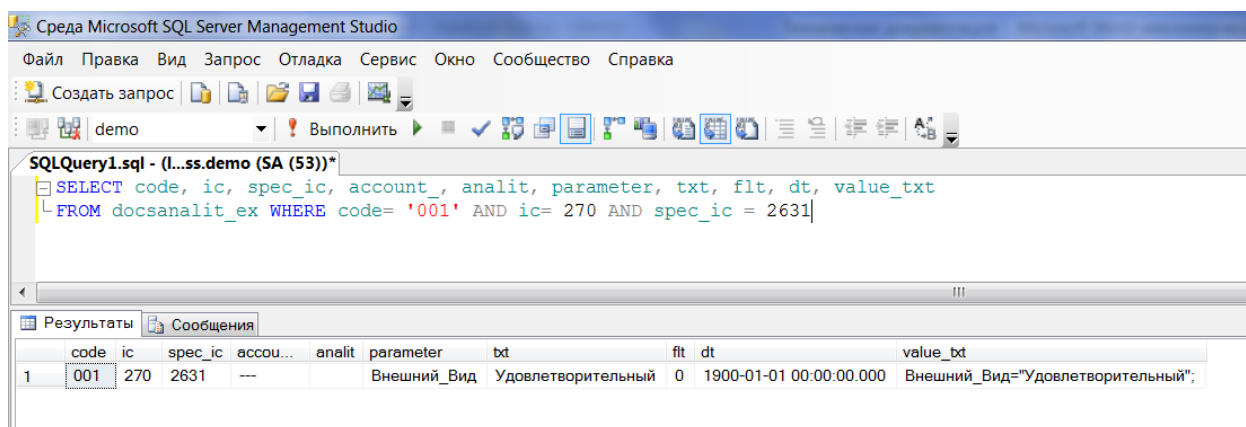
**Примечание.** Содержимое таблицы spr000 поддерживается структурой базы данных (триггерами, см. **стандартные триггеры таблиц шапки документов**) автоматически. Поэтому ручное изменение данных в этой таблице должно производиться аккуратно и только в случае крайней необходимости. Допускается «ручное» изменение признаков f\_sent (его установка блокирует удаление документа, но не пометку на удаление), f\_kkm, f\_print, f\_singlennline, f\_manualrate. Самостоятельное изменение признака f\_deleted и полей maindb, mainic - недопускается. Изменения остальных полей будет отменено первой корректировкой исходного документа.

### Дополнительные реквизиты документов, определенные в точке зрения.

Теперь выполним запрос для просмотра записей, созданных в таблице дополнительных реквизитов. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code), идентификатор документа (поле ic) и идентификатор строки документа, для которой

определен параметр (spec\_ic). Если поле spec\_ic содержит значение 0, то это означает что реквизит относится ко всему документу.

```
SELECT code, ic, spec_ic, account_, analit, parameter, txt, flt, dt, value_txt
FROM docsanalit_ex WHERE code= '001' AND ic= 270 AND spec_ic = 2631
```



Имя поля	Значение	Комментарий
code	001	Код типа документа.
ic	270	Идентификатор документа.
spec_ic	2631	Идентификатор строки документа.
account_	---	Код модели учета, к которой относится реквизит. Для реквизитов определенных в точке зрения (для всех учетов) в этом поле указано ---. Для реквизитов, определенных в бизнес-функции документа будет указан код модели учета из выбранной бизнес-функции.
parameter	Внешний_Вид	Наименование параметра, определяющего дополнительный реквизит в точке зрения.
analit		В данном примере значение этого поля игнорируется. Оно используется для реквизитов с типом Аналитика (выбор значения из справочника). В таком случае в него было бы записано выбранное значение (код аналитики и код из соответствующего справочника).
txt	Удовлетворительный	Значение реквизита. Так как наш параметр описывает простую текстовую строку, введенное значение сохраняется в этом поле.
flt	0	В данном примере значение этого поля игнорируется. Оно используется для реквизитов с типом Число вещественное, Число целое, Логическое.
dt	1900-01-01 00:00:00.000	В данном случае значение этого поля игнорируется. Оно используется для реквизитов с типом Дата и время.
value_txt	Внешний_Вид='Удовлетворительный';	Форма значения реквизита для загрузки в Вычислитель. Указанную строку можно передать вычислителю, и в нем определяются все необходимые переменные.

**Примечание.** Для упрощения доступа к дополнительным реквизитам документа в структуре базы данных Айтида предусмотрены 4 хранимые функции:

1. *fn\_getchar\_ex* ( @objtype char(3), @objcode char(3), @parameter varchar(250), @code varchar(20), @doc\_ic int= 0, @spec\_ic int = 0, @date datetime= '' ) – для получения текстового значения дополнительного реквизита.
2. *fn\_getint\_ex* ( @objtype char(3), @objcode char(3), @parameter varchar(250), @code varchar(20), @doc\_ic int= 0, @spec\_ic int = 0, @date datetime= '' ) – для получения целочисленного значения дополнительного реквизита.
3. *fn\_getfloat\_ex* ( @objtype char(3), @objcode char(3), @parameter varchar(250), @code varchar(20), @doc\_ic int= 0, @spec\_ic int = 0, @date datetime= '' ) – для получения вещественного значения дополнительного реквизита.
4. *fn\_getdatetime\_ex* ( @objtype char(3), @objcode char(3), @parameter varchar(250), @code varchar(20), @doc\_ic int= 0, @spec\_ic int = 0, @date datetime= '' ) – для получения значения дополнительного реквизита типа дата и время.



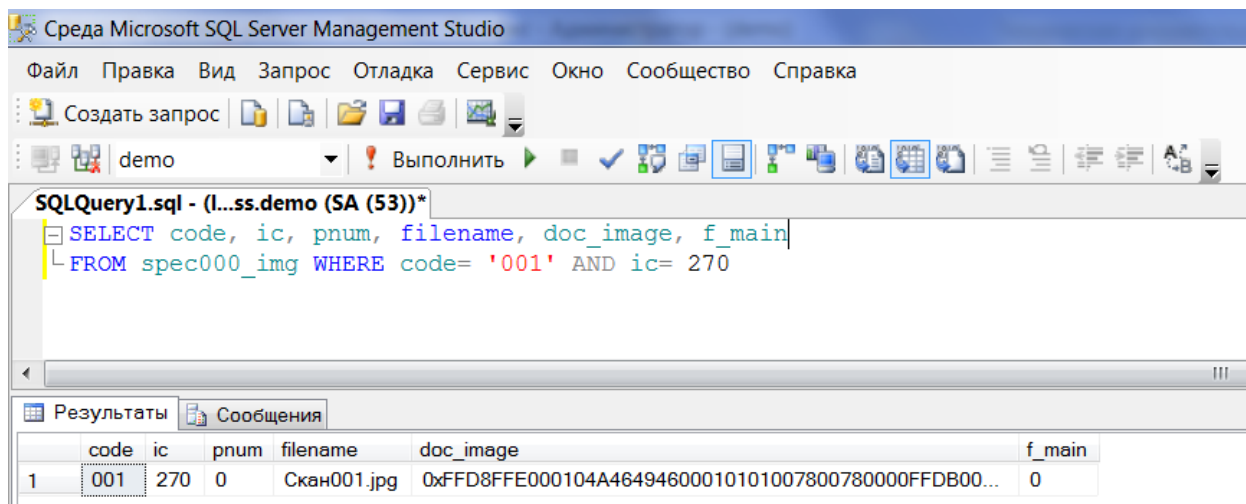
Пример использования этих функций:

```
SELECT ndok, date, dbo.fn_getchar_ex( 'DOC', '001', 'Внешний_Вуд', '', identity_column, 2631, '' ) FROM spr001 WHERE identity_column= 270
```

Дополнительные файлы, прикрепленные к документу.

Теперь выполним запрос для просмотра записей, созданных в таблице дополнительных файлов, прикрепленных к документу. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code) и идентификатор документа (поле ic).

```
SELECT code, ic, pnum, filename, doc_image, f_main  
FROM spec000_img WHERE code= '001' AND ic= 270
```

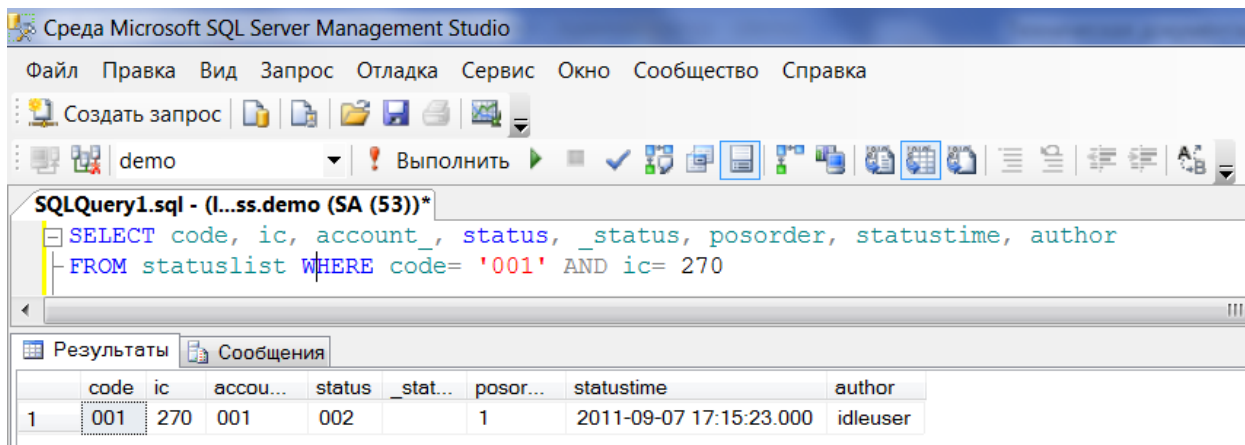


Имя поля	Значение	Комментарий
code	001	Код типа документа.
ic	270	Идентификатор документа.
pnum	0	Номер файла по порядку в списке прикрепленных файлов
filename	Скан001.jpg	Имя исходного файла.
doc_image	0xFFD8FFE000104A464...	Образ файла в двоичном виде
f_main	0	Пользовательский признак, которым можно отметить основной прикрепленный файл.

История изменения статусов документов.

Теперь выполним запрос для просмотра записей, созданных в таблице изменения статусов документов. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code) и идентификатор документа (поле ic).

```
SELECT code, ic, account_, status, _status, posorder, statustime, author  
FROM statuslist WHERE code= '001' AND ic= 270
```

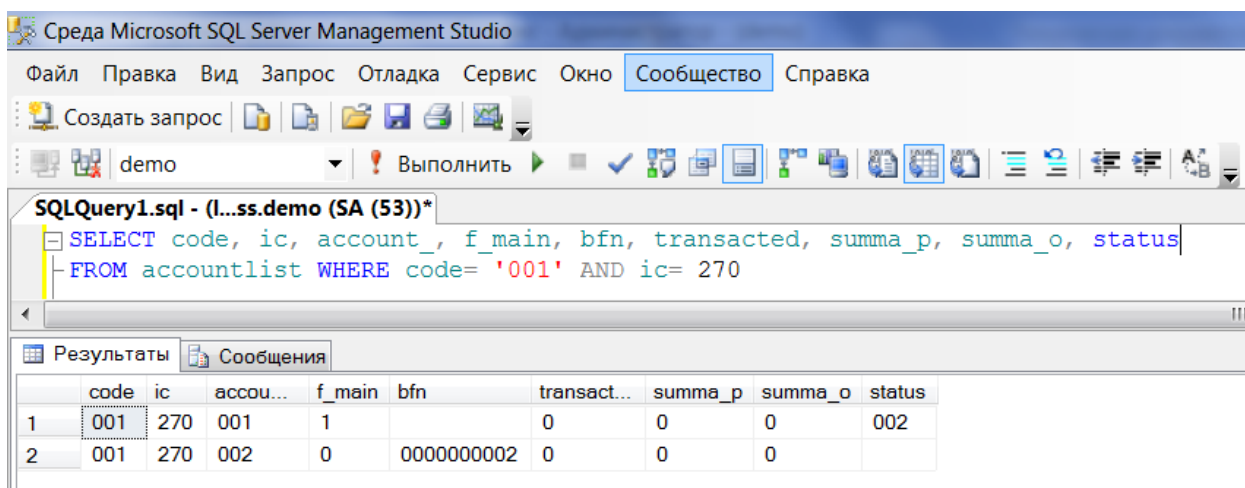


Имя поля	Значение	Комментарий
code	001	Код типа документа.
ic	270	Идентификатор документа.
account_	001	Модель учета, для которого определен присвоенный статус
status	002	Код статуса из справочника статусов документов
_status		Код статуса, который был до присвоения. В данном случае статус был присвоен при создании документа. Предыдущего не было.
posorder	1	Номер по порядку.
statustime	2011-09-07 17:15:23.120	Дата и время присвоения статуса
author	idleuser	Пользователь присвоивший статус документу. Поле устанавливается автоматически.

Список моделей учета, на которые должен влиять документ.

Теперь выполним запрос для просмотра записей, созданных в таблице моделей учета, на которые должен влиять документ. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code) и идентификатор документа (поле ic).

```
SELECT code, ic, account_, f_main, bfn, transacted, summa_p, summa_o, status
FROM accountlist WHERE code= '001' AND ic= 270
```

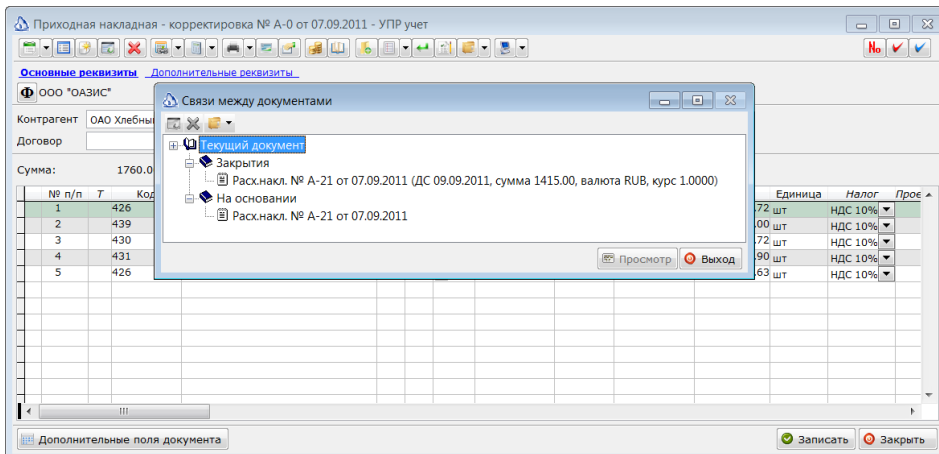
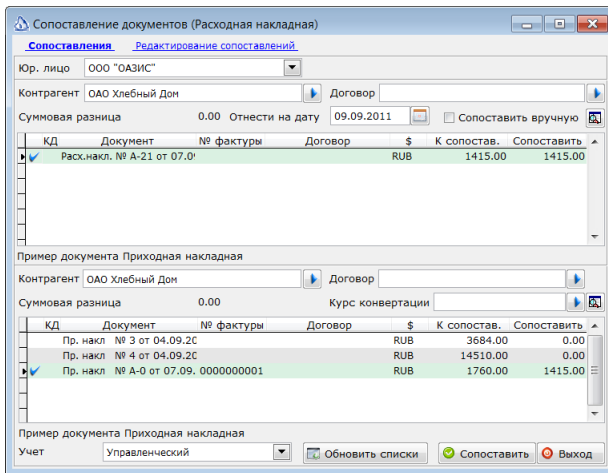


Имя поля	Значение	Комментарий
code	001	Код типа документа.
ic	270	Идентификатор документа.
account_	001	Модель учета, на которую должен влиять документ.
f_main	1	Признак того, что последний раз документ корректировался именно с этим учетом.
bfn		Код бизнес – функции, выполняемой документом в модели учета. Обратите внимание, в управленческом учете (account_ = 001), бизнес –

		функция не указан, а в бухгалтерском (account_ = 002) указана.
transacted	0	Признак принятия документа к учету. В данном примере ни к одному из учетов документ не принят.
summa_p	0	Общая сумма сопоставлений – оплата документа. В данном примере документ ни с какими другими документами не сопоставлен.
summa_o	0	Общая сумма сопоставлений – отгрузка/поставка документа. Приходная накладная не может быть дополнительно отгружена или поставлена, поэтому для нее это поле всегда равно 0.
status	002	Текущий (последний присвоенный) статус документа.

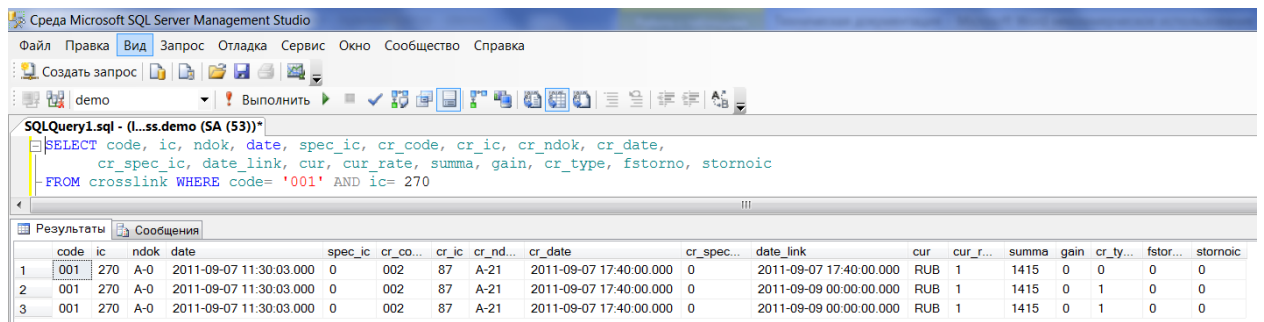
### Таблица связей между документами.

Для рассмотрения содержимого таблицы связей между документами создадим еще один документ – Расходная накладная. Создадим ее на основании рассмотренной приходной накладной, но с меньшей суммой и сопоставим эти два документа друг с другом. Обратите внимание, что в окне сопоставления документов указано поле «Отнести на дату», отличное от дат обоих документов. На последнем рисунке показаны связи между документами после сопоставления.



Для просмотра содержимого таблицы связей между документами (crosslink) выполним следующий запрос. Обратите внимание на условие отбора записей. Оно содержит код типа документа (поле code) и идентификатор документа (поле ic).

```
SELECT code, ic, ndok, date, spec_ic, cr_code, cr_ic, cr_ndok, cr_date,
       cr_spec_ic, date_link, cur, cur_rate, summa, gain, cr_type,
       fstorno, stornoic
FROM crosslink WHERE code= '001' AND ic= 270
```



Запрос вернул три строки. Первая из них, со значением в поле cr\_type = 0 соответствует связи «на основании», две остальных – сопоставление с расходной накладной. Записей по сопоставлению две, т.к. произведено сопоставление по двум учетам по управленческому и бухгалтерскому. Рассмотрим более подробно запись о сопоставлении по управленческому учету.

Имя поля	Значение	Комментарий
code	001	Код типа сопоставляемого документа.
ic	270	Идентификатор сопоставляемого документа.

account_	001	Модель учета, для которой производится сопоставление.
ndok	A-0	Номер сопоставляемого документа.
date	2011-09-07 11:30:03.000	Дата сопоставляемого документа.
spec_ic	0	Для документов, которые позволяют сопоставлять отдельные строки многострочной части (банковская выписка, авансовый отчет, внешнее движение средств), будет указан идентификатор строки, для всех остальных документов 0.
cr_code	002	Код типа сопоставленного документа.
cr_ic	87	Идентификатор сопоставленного документа.
cr_ndok	A-21	Номер сопоставленного документа.
cr_date	2011-09-07 17:40:00.000	Дата сопоставленного документа.
cr_spec_ic	0	Идентификатор строки сопоставленного документа (см. spec_ic).
date_link	2011-09-09 00:00:00.000	Дата связи. В данном примере она была указана в форме сопоставления в поле «Отнести на дату». По умолчанию, она была бы равна максимальной из дат сопоставляемых документов.
cur	RUB	Валюта сопоставляемого документа.
cur_rate	1	Курс конвертации валюты сопоставляемого документа.
summa	1415	Сумма сопоставления. Она может отличаться от суммы документа, но не может превышать ее. Сумма по всем записям его сопоставлений не может превышать суммы документа.
gain	0	Суммовая разница, рассчитанная для данного сопоставления. Равна разнице в курсе документа и курсу системы на дату отнесения умноженной на сумму сопоставления.
cr_type	1	Тип связи. В данном случае 1 – сопоставление – оплата.
fstorno	0	Признак сторнирования предшествующего сопоставления. В данном примере это первичное сопоставление.
stornoic	0	Идентификатор записи сопоставления в случае сторнирования.

## Справочники

Справочник - средство для ведения списков однородных элементов данных. Помимо наименования элементов данных списки могут содержать различную дополнительную информацию. При помощи справочников организуется ввод стандартной информации в документы, а также ее просмотр и, если необходимо, корректировка. В большинстве своем справочники являются электронными аналогами каталогов. Каждая карточка - это строка справочника, а сведения, заносимые в карточку, являются реквизитами справочника. Например. Для того чтобы покупатель, продавец, кладовщик, директор однозначно понимали, о каком товаре идет речь, каждый должен называть его одинаково, т.е. в соответствии с однажды утвержденным справочником товаров.

Справочники в системе Айтида делятся на две группы:

1. Первая группа служит для сбора и классификации различной информации необходимой для ведения финансово-хозяйственной деятельности предприятия.
2. Вторая необходима для настройки работы самой системы Айтида. Все настройки системы Айтида производятся через заполнение соответствующих справочников.

В отличие от документов, справочники не влияют непосредственно на модели учетов. Набор реквизитов для каждого справочника является уникальным, т.к. каждый справочник содержит реквизиты для описания конкретного объекта.

Все справочники системы Айтида обязательно содержат как минимум два реквизита – код (поле code) и наименование (поле name). Единственным исключением из этого правила является справочник товаров. В нем внутренний код в системе содержится в поле np, а поле code является неуникальным артикулом товара. В некоторых случаях, например, в случае справочника товаров, описание одного объекта, товара, разнесено на несколько связанных между собой таблиц. В таком случае все взаимосвязи между таблицами осуществляются по коду (code) объекта.

В системе Айтида применяется общее правило использования ссылок на объекты справочников – по коду объекта. Этим справочники отличаются от документов, для которых все ссылки осуществляются по идентификатору документа (поле identity\_column таблицы шапки документа).

В системе Айтида все справочники могут иметь иерархическую структуру. Это означает, что элементы справочника могут быть разделены на группы и помещены в папки. Папки могут вложенными в другие папки. Система Айтида не ограничивает глубину вложения папок. Предусмотрено два способа описания иерархии (папок).

1. Если справочник не добавлен в Точку Зрения, то папки справочника содержатся в самой таблице справочника. В этом случае используется одно поле name для описания наименования папки. Поле folder устанавливается в значение 1. В такой ситуации все вложенные в папку элементы (карточки и другие папки) используют поле parent для указания своей родительской папки. Поле parent ссылается на уникальный идентификатор записи (identity\_column) папки.
2. Для справочников, добавленных в точку зрения необходимо наличие внешней структуры папок, которая описывается в справочнике структур. В этом случае, папки не хранятся в таблице справочника. Поле parent в таблице справочника не используется. Все папки и вложенные в них элементы хранятся в таблице spectree.

В системе Айтида применяется единое правило именования реквизитов в структуре таблиц объектов. Если реквизиты разных объектов содержат однородную информацию, например, код контрагента, то имя и тип поля в соответствующих таблицах будут совпадать. Ниже приведены обязательные для всех справочников реквизиты.

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
code	char(3, 7, 10, 20)	Уникальный код записи в справочнике. Во всех справочниках по полю code созданы уникальные индексы, не позволяющие появляться дубликатам записей. В различных справочниках используются разные длины кодов – 3 символа (например, справочник складов), 7 символов (например, справочник контрагентов), 10 символов (например, справочник товаров), 20 символов (справочник расчетных счетов). Данное поле недоступно для ввода пользователями. Оно генерируется автоматически в момент создания новой карточки справочника.
name	varchar(250)	Наименование объекта, описываемого в карточке справочника. Данное поле не является уникальным и вводится пользователями в карточке.
parent		Идентификатор записи родительской папки, в случае хранения структуры папок в таблице справочника. В противном случае – 0
notavail		Признак, того, что элемент справочника перенесен в архив и не должен больше использоваться.
folder		Признак того, что запись содержит папку элементов, а не конечный элемент справочника. Данное поле используется в случае хранения структуры папок в таблице справочника.

Полный набор реквизитов всех справочников можно найти в приложении №...

#### Типы справочников системы Айтида.

В системе Айтида определено достаточно большое количество справочников, которые позволяют хранить информацию о большинстве используемых и учитываемых объектов. Тем не менее, при возникновении ситуации, когда необходимо добавить в систему новый объект учета и, соответственно, новый справочник, система Айтида позволяет это сделать в рамках настройки без изменения структуры базы данных. Процесс добавления и использования новых справочников будет описан далее. Здесь приведен список predetermined справочников, с указанием таблиц для хранения реквизитов и уникальных ключей. Данный список хранится в служебной таблице `sprsystemtabs`, и может быть использован для получения информации о справочнике. Аналогичная информация о новых справочниках, определённых в процессе настройки системы хранится в таблице `sprlocaltabs`.

Код типа справочника	Наименование справочника	Основная Таблица/представление справочника	Основная таблица справочника	Ключевое выражение справочника	Ключевое выражение таблицы справочника	Выражение, определяющее тип справочника в

						<b>основной таблице</b>
S01	Справочник контрагентов	sprclient	sprclient	code	code	cat='001'
S02	Справочник ТМЦ	sprnn	sprres	nn	code	cat='001'
S04	Справочник понятий	sprconcepts	sprconcepts	code	code	
S05	Справочник работ и услуг	sprservice	sprres	code	code	cat='002'
S06	Справочник сотрудников	sprempl	sprempl	code	code	
S07	Справочник бизнес функций	sprbfm	sprbfm	code	code	
S09	Справочник центров ответственности	sprmvz	sprmvz	code	code	
S10	Справочник ресурсов	sprres	sprres	code	code	cat='100'
S14	Справочник расширенных характеристик	sprdim	sprdim	code	code	
S15	Справочник групп ресурсов	sprres_g	sprres_g	code	code	
S16	Справочник категорий ресурсов	sprres_c	sprres_c	code	code	
S19	Справочник единиц измерения	spredn	spredn	code	code	
S1B	Справочник окружений	sprenv	sprenv	code	code	
S1D	Справочник направлений деятельности	spractivity	spractivity	code	code	
S1E	Справочник проектов	sprprojects	sprprojects	code	code	
S1F	Справочник параметров и полей ввода	sprparams	sprparams	code	code	
S21	Справочник отчетных форм	qreport_templates	qreport_templates	code	code	
S22	Справочник импорта/экспорта	sprimpex	sprimpex	code	code	
S29	Справочник основных средств	spros	sprres	code	code	cat='003'
S30	Справочник организаций	sprfirm	sprfirm	code	code	
S31	Справочник валют	sprcur	sprcur	code	code	
S32	Справочник складов	sprskl	sprskl	code	code	
S33	Справочник кодов кассы и р с	sprkskod	sprkskod	code	code	
S35	Справочник кодов налогов	sprkodn	sprkodn	code	code	
S36	Справочник филиалов	sprbranch	sprbranch	code	code	
S37	План счетов	sprbs	sprbs	bs+ss+ss2+ss3+level –	bs+ss+ss2+s3+level_	
S38	Справочник типов аналитики	spranalit	spranalit	code	code	
S39	Справочник затрат	sprztr	sprztr	code+subcode	code+subcode	



S3F	Справочник групп пользователей	usergroups	usergroups	code	code	
S40	Справочник списков налогов	sprlistkodn	sprlistkodn	code	code	
S41	Справочник банков	sprbank	sprbank	code	code	
S42	Справочник р/с наших фирм	sprrsfirm	sprrsfirm	firm+rs	firm+rs	
S43	Справочник расчетных счетов	sprrs	sprrs	code	code	
S44	Справочник показателей	sprbzo	sprbzo	bzo+so+so2+so3+level_ el_	bzo+so+so2+so3+level_ el_	
S45	Справочник стран	sprcountry	sprcountry	code	code	
S47	Справочник функций пользователя	sprfunctions	sprfunctions	code	code	
S48	Справочник производителей	sprmanuf	sprmanuf	code	code	
S49	Справочник норм амортизации	sprnamortos	sprnamortos	code	code	
S4A	Справочник типов табелей учета рабочего времени	sprwtabletypes	sprwtabletypes	code	code	
S4B	Справочник групп расчета цен	sprccgroup	sprccgroup	code	code	
S4C	Справочник алгоритмов округления при расчете цен	sprccround	sprccround	code	code	
S4D	Справочник торгового оборудования	sprequipment	sprequipment	code	code	
S50	Справочник групп по ОКОФ	sprokof	sprokof	code	code	
S51	Справочник групп ОС	sprosgroups	sprosgroups	code	code	
S52	Справочник операций ОС	sprosoper	sprosoper	code	code	
S53	Справочник статей доходов	sprprofit	sprprofit	code+subcode	code+subcode	
S54	Справочник групп контрагентов	sprclientgroup	sprclientgroup	code	code	
S56	Справочник журналов операций	sprbsoper	sprbsoper	code	code	
S57	Справочник планов счетов	sprbsspr	sprbsspr	code	code	
S58	Справочник депозитариев	sprdeposit	sprdeposit	code	code	
S59	Справочник шаблонов справочников	sprsprtemplate	sprsprtemplate	code	code	
S5A	Справочник вариантов меню	sprmenu	sprmenu	code	code	
S5D	Справочник видов товаров (алкогольной продукции)	sprnnvid	sprnnvid	code	code	
S5E	Справочник договоров	sprcontract	sprcontract	code	code	
S5F	Справочник групп счетов	sprinvoicegroups	sprinvoicegroups	code	code	
S66	Справочник кодов явок/неявок	sprwstate	sprwstate	code	code	

S67	Табель учета рабочего времени	sprwtable	sprwtable	code	code	
S6A	Справочник карт клиентов	sprmcards	sprmcards	code	code	
S6B	Справочник скидок клиентам	sprmdiscount	sprmdiscount	code	code	
S6C	Справочник видов оплат	sprvpay	sprvpay	code	code	
S6D	Справочник владельцев карт (клиентов)	sprmcclient	sprmcclient	code	code	
S6E	Справочник схем скидок клиентам	sprmdiscount_schem	sprmdiscount_schem	code	code	
S6F	Справочник колонок прайс листа (категорий цен)	sprccnames	sprccnames	clev	clev	
S70	Справочник контекстов точек зрения	sprcontext	sprcontext	code	code	
S72	Справочник типов контрагентов	sprclienttypes	sprclienttypes	code	code	
S75	Справочник структур справочников	sprtrees	sprtrees	code	code	
S76	Справочник шаблонов хоз. операций	sprdocsbs	sprdocsbs	code	code	
S77	Справочник шаблонов бизнес нот	sprdocsbzo	sprdocsbzo	code	code	
S78	Справочник шаблонов правил проведения	sprdocsur	sprdocsur	code	code	
S79	Справочник типовых договоров	sprcontracttype	sprcontracttype	code	code	
S7A	Справочник сезонов	sprseasons	sprseasons	code	code	
S7B	Справочник норм естественной убыли	sprloss	sprloss	code	code	
S7C	Справочник журналов документов	sprjournal	sprjournal	code	code	
S7D	Справочники пользователей	sprxxx	sprxxx	sprcode+code	sprcode+code	
S7E	Справочник групп филиалов	sprbranchgroups	sprbranchgroups	code	code	
S7F	Справочник шаблонов правил обмена данными	sprexchange	sprexchange	code	code	
S80	Справочник выражений для расчетов и фильтров	sprexpressions	sprexpressions	code	code	
S82	Справочник для заполнения персональных сведений	sprsimple	sprsimple	code	code	
S83	Справочник категорий персональных сведений	sprsimplecat	sprsimplecat	code	code	
S84	Справочник статусов документов	sprstatus	sprstatus	code	code	
S8A	Справочник функций пользователей (обновленный)	qreport_functions	qreport_functions	code	code	
S8B	Справочник оборудования торгового зала	sprexequipment	sprexequipment	code	code	
S8C	Справочник графических и текстовых ресурсов системы	sprresource	sprresource	code	code	
S8E	Справочник заданий на обработку данных	sprschedule	sprschedule	code	code	

S8F	Справочник панелей инструментов	sprtoolbars	sprtoolbars	code	code	
S90	Справочник констант системы	sprconstants	sprconstants	code	code	
S91	Справочник праздничных дней	sprholidays	sprholidays	code	code	
S92	Справочник календарей	sprcalendar	sprcalendar	code	code	
S93	Список пользователей	username	username	code	code	
S98	Справочник моделей учетов	spraccount	sprsystemaccount	code	code	f_active=1
S99	Справочник точек зрения	sprviewpoints	sprviewpoints	code	code	
S9A	Справочник скидок	sprdiscounts	sprdiscounts	code	code	
S9B	Справочник вариантов доставки данных	sprexchangeservice	sprexchangeservice	code	code	
S9C	Справочник конфигураций ТСД	sprtsdconfigurations	sprtsdconfigurations	code	code	
S9D	Справочник мест нахождения	sprmnh	sprmnh	code	code	
S9E	Справочник профилей оборудования	spreqprofiles	spreqprofiles	code	code	
S9F	Справочник системных профилей	sprsystemprofiles	sprsystemprofiles	code	code	
SA0	Справочник фильтров для журналов документов и справочников	sprfilters	sprfilters	code	code	

**Примечание.** Некоторые справочники находятся в одной физической таблице с другими справочниками, например, справочник товаров и справочник услуг. В таком случае в колонке «**Основная Таблица/представление справочника**» указано имя представления, выборка из которого определяет состав справочника. Имя физической таблицы, указано в колонке «**Основная таблица справочника**». Структура таблицы и представления могут отличаться. Добавление записей для справочника должно осуществляться в основную таблицу с обязательным указанием значения полей указанных в колонке «**Выражение, определяющее тип справочника в основной таблице**»

### Дополнительные реквизиты справочников.

Достаточно часто, при реализации конкретных проектов, возникает ситуация, когда в каком либо справочнике необходимо сохранить дополнительную информацию, для которой нет стандартного реквизита. Для реализации требуемой возможности, системе Айтида предусмотрен механизм добавления новых реквизитов в справочники. Добавление новых реквизитов производится без изменения структуры таблиц базы данных. Значения всех новых реквизитов записываются в специальную таблицу – `spranalit_ex`. Дополнительные реквизиты справочников могут быть постоянными во времени и изменяемыми по времени – периодическими. Периодически реквизиты позволяют вводить своё значение с указанием даты, с которой должно использоваться данное значение. Описание структуры таблицы `spranalit_ex`:

Имя поля и тип данных	Назначение
sprcode char(3)	Код типа справочника, см. таблицу <code>sprtabs</code> . По этому полю определяется, тип справочника, к которому относится реквизит.
code char(20)	Код карточки справочника, к которой относится реквизит.
parameter varchar(250)	Имя параметра, описывающего дополнительный реквизит.
analit char(13)	Значение реквизита, в случае, если выбор производился из справочника. Первые три символа содержат код из справочника аналитик ( <code>spranalit</code> ), последующие символы содержат

	код выбранной карточки используемого справочника.
txt varchar(250)	Значение реквизита, в случае ввода текстового поля.
flt float	Значение реквизита, в случае ввода числового или логического значения.
dt datetime	Значение реквизита, в случае ввода значения типа дата и время.
f_nf bit	Признак, если 1, то при записи карточки справочника значение реквизита обязательно должно быть указано.
f_vp bit	Признак, того что дополнительный реквизит был добавлен в точке зрения. Для справочника товаров возможно добавление дополнительных реквизитов в справочнике <b>групп ресурсов</b> .
date datetime	При использовании <i>периодических</i> реквизитов, данное поле содержит дату, с которой начинает действовать указанное значение. Используется только, если поле f_periodic= 1
f_periodic bit	Признак, указывающий, что данный дополнительный реквизит является <i>периодическим</i> реквизитом.
value_txt text	Текстовое представление значения реквизита, которое можно передавать вычислителю. Например, для дополнительного реквизита с именем ДополнительныйРеквизит и значением в поле analit = " cln0010076", в поле value_txt будет указано: <b>ДополнительныйРеквизит_список="0010076";ДополнительныйРеквизит_папки="0";ДополнительныйРеквизит="0010076";ДополнительныйРеквизит_имя="ООО Чемоданчик";</b> Эту строку можно передать вычислителю, для определения соответствующих переменных с указанными значениями.

**Примечание.** Поля таблицы spranalit\_ex: ic, t\_concept, value\_ больше не используются и сохранены для обратной совместимости с более ранними версиями системы.

### Дополнительные справочники, определенные при настройке системы.

Система Айтида позволяет определить в системе новые справочники в процессе настройки. Данная настройка производится в справочнике "Справочники пользователей". При этом описание нового справочника сохраняется в таблицах sprsprtemplate и specsprtemplate. Информация, вводимая в новые справочники, хранится в таблице sprgxxx. Для удобства работы, для каждого нового справочника создается специальное представление, имя которого задается при настройке. В следующей таблице описана структура таблицы sprgxxx.

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
sprcode	char(3)	Код типа справочника. По этому полю происходит разделение информации в таблице на разные типы справочников
code	char(10)	Код карточки определённого справочника
name	varchar(250)	Наименование карточки справочника
ex_code	varchar(50)	Дополнительные код карточки справочника
fields	varchar(7500)	Поле, содержащее все описанные для справочника поля, кроме кода, наименования и дополнительного кода.
notavail	bit	Признак того, что карточка перенесена в архив и не должна использоваться.
folder	bit	Признак того, что запись содержит папку элементов, а не конечный элемент справочника. Данное поле используется в случае хранения структуры папок в таблице справочника.

**Примечание.** Для каждого поля справочника в поле fields выделяется участок, в который записывается значение этого поля. Представление, которое создается при настройке справочника, формирует список полей, преобразуя значения, взятые из этого поля к требуемому типу данных. Поэтому, при переопределении полей данных в описании справочника или при изменении их порядка, данные, которые уже находятся в базе, могут потерять свою актуальность. Пример представления для справочника Фронтал виды оплат:

```

CREATE VIEW sprview___38R0Q7KEB AS SELECT
    code, name,
    SUBSTRING(fields, 1, 250) AS ФронтолТекст,
    SUBSTRING(fields, 251, 3) AS ВидОплатРазрешенаСдача,
    ISNULL(CONVERT(int, SUBSTRING(fields, 254, 15)),0) AS ВидОплатТипОплаты,
    SUBSTRING(fields, 269, 3) AS ВидОплатИзменяетДенежныйЯщик,
    SUBSTRING(fields, 272, 3) AS ВидОплатОперация,
    ISNULL(CONVERT(int, SUBSTRING(fields, 275, 15)),0) AS ВидОплатКодПлатежнойСистемы,
    SUBSTRING(fields, 290, 10) AS ВидОплатКодГруппыСертификатов,
    ISNULL(CONVERT(int, SUBSTRING(fields, 300, 15)),0) AS ВидОплатКодГруппыПечати,
    identity_column, updated, received, edit_dat, timestamp_column, farc, posorder,
    caccess, taccess, folder, parent, notavail
FROM sprxxx WHERE sprcode= '018'

```

### Таблица **sprmain**.

Так же, как для документов определена таблица **spr000** – журнал всех документов, для справочников предусмотрена таблица **sprmain**. В этой таблице копится информация (значения полей, присутствующих во всех или многих справочниках) из всех справочников системы. Добавляется информация автоматически при добавлении записей в соответствующий справочник. В дальнейшем, эта информация обновляется по мере изменения исходных справочников, но никогда не удаляется, даже после удаления исходных карточек справочников. Кроме того, в этой таблице сохраняется дата и время удаления исходной карточки, а так же пользователь, выполнивший удаление.

Так же таблица **sprmain** играет важную роль, при синхронизации карточек справочников в процессе обмена данными между филиалами. В ней сохраняется информация о филиале, в котором первоначально была создана карточка и идентификатор этой карточки в исходном филиале.

В следующей таблице приведена структура **sprmain**.

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
scode	char(3)	Код типа справочника. По этому полю происходит разделение информации в таблице на разные типы справочников
ic	int	Идентификатор карточки в основной таблице справочника. Данное поле очищается при удалении записи из основной таблицы. Поэтому, при поиске информации по коду (поле code) необходимо проверять поле ic, чтобы оно не имело значения 0.
code	char(20)	Код карточки в основной таблице справочника.
icode	int	Целочисленное представление кода карточки, для вычисления кода при создании новой карточки.
maincode	char(10)	Поле maincode, для справочников, в которых есть это поле. Основное использование поля maincode – в справочниках товаров и услуг.
cat	char(3)	Поле cat, для справочников, в которых есть это поле. Основное использование – деление единого справочника на несколько подкатегорий (например, справочник ресурсов делится на товары, услуги, ОС и т.п.)
ex_code	varchar(50)	Поле Дополнительный код для справочников, в которых есть это поле.
namr	varchar(250)	Наименование карточки справочника.

parent	int	Поле parent в основной таблице справочника.
folder	bit	Поле folder в основной таблице справочника.
maindb	char(3)	Код филиала, в котором была изначально создана карточка.
mainic	int	Идентификатор записи в основной таблице справочника в филиале, в котором изначально была создана карточка
notavail	bit	Поле parent в основной таблице справочника.
author	varchar(128)	Пользователь, создавший карточку справочника.
created	datetime	Дата и время создания карточки справочника.
editor	varchar(128)	Пользователь, последний изменивший карточку справочника. Более подробная история изменений хранится в таблице <a href="#">sprauthor</a> .
deleter	varchar(128)	Пользователь, удаливший карточку из основной таблицы справочника
deleted	datetime	Дата и время удаления карточки справочника.
f_sent	bit	Признак, что карточка справочника была передана в другой филиал. Данное поле проверяется при попытке удаления карточки справочника.

**Примечания.** Данная таблица заполняется и поддерживается в актуальном состоянии автоматически триггерами основных таблиц справочников (см. **стандартные триггеры основных таблиц справочников**). Поэтому ручное изменение данных в этой таблице должно производиться аккуратно и только в случае крайней необходимости. Допускается «ручное» изменение признака f\_sent (его установка блокирует удаление карточки справочника, но не перенос карточки в архив). Самостоятельное изменение полей maindb, mainic – недопускается, т.к. это приведет к невозможности производить обмен данными между филиалами. Изменения остальных полей будет отменено первой корректировкой исходной карточки справочника.

### Иерархия в справочниках.

Как уже упоминалось в системе Айтида предусмотрено два варианта хранения иерархии элементов справочников.

1. Для справочников, которые не включены в точку зрения, иерархия хранится в основной таблице справочника.
2. Для справочников, которые включены в точку зрения, должна быть определена по крайней мере одна иерархия (структура) в справочнике [структур](#).

Общий принцип хранения иерархии заключается в хранении идентификатора родительской папки в карточке справочника. В первом случае это делается в основной таблице справочника, во втором – в таблице [spectree](#).

Исходя из указанного принципа видно, что иерархия может иметь любую глубину и, в случаях ошибок, может иметь циклические ссылки. Поэтому при формировании дерева папок всегда необходимо ограничивать глубину выборки или проверять папки на цикличность.

Для упрощения решения задачи формирования структуры справочника, в основной для отчетов или экспорта данных, предусмотрена хранимая процедура:

```
sp_gettree @sprcode char(3), @topparent int, @stru varchar(10), @currentLevel int, @tmpTable varchar(30),
@notNeedCreateTMP int = 0, @onlyAvail int = 0, @needChild int = 0
```

Наименование параметра	Имя параметра	Назначение
------------------------	---------------	------------

@sprcode	char(3)	Код типа справочника, для которого необходимо сформировать структуру.
@topparent	int	Папка, с которой надо начинать формирование. Если требуется сформировать структуру полностью, то указывается значение 0
@stru	char(10)	Код структуры из справочника структур. Если в качестве параметра указано пустое значение, то структура будет сформирована по основной таблице справочника.
@currentLevel	int	Внутренние счетчик рекурсии, т.к. процедура вызывается рекурсивно, а глубина рекурсии ограничена, то максимальная глубина папок, которую может вернуть данная процедура, равна 32.
@tmpTable	varchar(30)	Перед вызовом процедуры можно сформировать временную таблицу, в которую будет записан результат. Структура таблицы должна быть следующая: <i>code char(10), name varchar(250), ic int, parent int, parentname varchar(250), folder bit, identity_column int NOT NULL IDENTITY(1,1)</i> После заполнения выборка из этой таблицы должна производиться в порядке <i>identity_colum – SELECT * FROM (ИмяВременнойТаблицы) ORDER BY identity_column</i>
@notNeedCreateTMP	int	Значение 1 должно указываться только с непустым именем временной таблицы.
@onlyAvail	int	Если указано значение 1, то будут выбраны только карточки, которые не перенесены в архив. Иначе – все карточки.
@needChild	int	Параметр используется только для справочника "S02", справочник товаров. Если указано значение 1, то будут выгружены и "заголовочные" товары и товары-характеристики. Иначе, товары – характеристики не попадут в выборку.

Для определения наиболее подходящей структуры для текущего пользователя или указанной точки зрения необходимо использовать хранимую функцию

*fn\_getsuitabletree ( @viewpoint char(10), @predefContext char(3), @objtype char(3), @objcode char(3))*

Наименование параметра	Имя параметра	Назначение
@viewpoint	char(10)	Код точки зрения, из которой необходимо извлекать информацию об используемой структуре. Если указан пустой код точки зрения, то будет использована точка зрения указанная «по умолчанию» для группы пользователей, в которую входит текущий пользователь.
@predefContext	char(3)	Контекст, для которого определяется структура. Допустимые значения: _MN – вызов и системы меню, RPT – вызов из отчета, 'S' + Допустимый код типа справочника, например 'SS02', 'D' + Допустимый код типа документа, например 'D001'.
@objtype	char(3)	Тип объекта, для которого определяется структура. Допустимые значения для этого параметра – 'SPR' – для справочника, 'DOC' – для документа.
@objcode	char(3)	Код типа справочника или документа, для которого определяется структура.

### Справочник аналитик.

В различных настройках системы Айтида можно указывать, что пользователь выбирает значение из указанного справочника и это значение сохраняется для дальнейших расчетов или как информационный реквизит документа или карточки справочника. Типичным примером таких настроек являются дополнительные реквизиты документов и справочников или аналитики в проводках хозяйственных операций. Для сохранения введенной информации недостаточно сохранить выбранный или вычисленный код, необходимо сохранить привязку этого кода к

справочнику, из которого была произведена выборка. Кроме того, существуют ситуации, когда один и тот же справочник может “возвращать” в систему значения для разных реквизитов. Например, справочник сотрудников организаций, может быть использован как источник данных для Сотрудника, для Менеджера или для МОЛ (материально ответственного лица). Для того чтобы, например, в проводках была возможность различать различные варианты использования одного справочника используется справочник аналитик. Основная таблица справочника аналитик – spranlit.

Наименование реквизита	Имя поля и тип данных в таблице документа	Назначение
code	char(3)	Код аналитики. Значение этого поля используется как первые три символа при сохранении значений аналитик в системе.
name	varchar(250)	Наименование аналитики.
sprcode	char(3)	Код справочника, являющегося источником значений для аналитики.
t_key	varchar(250)	Ключевое выражение для справочника. Аналогично полю t_key в таблице sprtabs
t_condept	varchar(250)	Базовое понятие, для определения имени поля контекста, из которого необходимо брать значение при формировании проводок и прочих вычислениях.
t_concept_k	varchar(250)	Базовое понятие, для определения имени поля контекста, из которого необходимо брать значение при формировании значения аналитики для кредитовой части проводки. Например, аналитика Склады, для склада отправителя это будет поле sklad, для склада получателя – sklad_d.
default_	varchar(250)	Значение по умолчанию. Используется, если не удалось вычислить значение аналитики.

### Характеристики товаров.

В системе Айтида реализован механизм дополнительных характеристик товаров, который позволяет к группе ресурсов, привязать набор характеристик. При этом для товаров, входящих в такую группу, будет доступна возможность создания подчиненных карточек товаров с указанием значений для каждой выбранной характеристики. Для реализации данного механизма предусмотрены следующие справочники:

1. Справочник характеристик (sprdim). В этом справочнике хранится список всех допустимых характеристик. Характеристика, помимо кода и наименования, характеризуется типом принимаемого значения и единицей измерения своего значения. Например, на складе могут храниться доски с различными значениями характеристики Длина, которая измеряется в метрах.
2. Справочник аналитик. Если характеристика использует справочник, в качестве источника своих значений, то связь с соответствующим справочником осуществляется через справочник аналитик.
3. Справочник комбинаций характеристик (sprdimcomb). В этом справочнике хранятся комбинации введенных в карточки товаров характеристик. По этому справочнику производится проверка на наличие вновь вводимой комбинации в уже существующих. Значения характеристик хранятся в таблице specdimcomb.

Для создания новой характеристики использующей в качестве источника своих значений новый справочник, например Фасон обуви, необходимо создать такой справочник в справочниках



пользователей, создать аналитику, использующую новый справочник. Вновь созданную аналитику можно указать в качестве типа значения для новой характеристики.

Для упрощения выше описанных действий, в интерфейсе системы дана возможность, пользователю нажав одну кнопку, и указав название нового справочника, создать все необходимые объекты. При этом будет создан новый справочник с двумя полями для ввода – *Код* и *Наименование*. При попытке повторно создать справочник с таким же именем, система заблокирует попытку и выдаст соответствующее предупреждение. Проверка на наличие указанного наименования производится в **двух(!)** справочниках – *справочнике аналитик* и *справочниках пользователя*. Поэтому, для повторного создания характеристики с уже использованным названием, необходимо сначала удалить ранее созданную аналитику и справочник пользователя.

Для получения значения конкретной характеристики товара необходимо использовать функцию `fn_getnndim( @nn char(10), @dim char(10) )`, где `@nn` – внутренний код карточки товара-характеристики, `@dim` – код характеристики. Для получения списка внутренних кодов товаров с указанной характеристикой необходимо использовать функцию `fn_getnnwithdim( @nn char(10), @dim char(10), @value_ varchar( 250 ) )`, где `@nn` – внутренний код заголовочного товара (товара, к которому “привязаны” товары – характеристики), `@dim` – код характеристики, `@value_` - значение характеристики. Следует иметь в виду, что для характеристик с типом *аналитика* значение в первых трех символах должно содержать код аналитики. Примеры использования данных функций будут рассмотрены ниже.

### Пример заполнения таблиц справочника.

Рассмотрим на примере заполнения карточки товара, какая информация записывается в описанные таблицы. Для этого в справочнике товаров создадим новую карточку и заполним ее следующим образом. В примере будет заполнено максимально возможное количество полей, поэтому содержимое некоторых полей будет не всегда согласовано между собой.

The image displays two screenshots of the 'Карточка ТМЦ' (Inventory Card) form in a software application. The left screenshot shows the 'Основные реквизиты' (Basic requisites) tab, and the right screenshot shows the 'Цены, комплектация и доп. информация' (Prices, configuration and additional information) tab.

**Left Screenshot (Basic requisites):**

- Код в гл. базе: 123456
- Доп. код: 000012/MAIN
- Наименование: Основная карточка товара
- Кр. наименование: Основная карточка товара
- Артикул: 121000-17
- Объем: 12.500 миллилитр
- Группа товара: Одежда
- Группа стат.: Алкоголь
- Бух-кая группа: не выбрана
- Вес: 5.000 грамм
- ЕИ: штука
- Кол-во в упак.: 20.0000
- Производитель: ЗАО "Метшвейстрой"
- Страна: РОССИЯ
- Налоги: НДС 18%
- № ГТД: 10126010/310810/0013370
- Срок хранения: 15 (дней)
- Штриховые коды: 4690501005039

**Right Screenshot (Prices, configuration and additional information):**

- Основная цена: 0.00
- Процент наценки: 15.000 %
- Группа расчета цен: Основная группа
- Мин. цена продажи: 350.00
- Максимальная скидка: 15.00
- Минимальный остаток: 10.000
- Максимальный остаток: 30.000
- Поставщик: 000 Закупка
- Импортер/Производитель: [empty]
- Дополнительная информация: [empty]
- Комментарии: [empty]
- Дополнительно:
  - Характеристики: [empty]
  - Единицы измерения: [empty]
  - Альтернативные замены: [empty]
  - Реквизиты по филиалам: [empty]
  - Печать штрих-кода обязательна: [empty]
- Комплектация:
  - Может быть собран: [checked]
  - Собирать автоматически: [empty]
  - Может быть разобран: [empty]
  - Использовать автоматические замены: [empty]
  - Список существующих калькуляций: [empty]

**Карточка ТМЦ**

Основные реквизиты | Цены, комплектация и доп. информация | **Алкоголь, нормы, скидки** | Дополнительные свойства

Крепость: 18,000 | Крепость поставщика: 18,500  
 Код в базе поставщика: F0000028084  
 Тип упаковки: Пакет  
 Вид продукции: не выбран

**Нормы естественной убыли**

При реализации: прочие весовые бакалейные товары | 5,000  
 При хранении: прочие весовые кондитерские товары | 7,000

**Скидка**

Скидка: Скидка 3 %

В архиве |

**Коды ТМЦ для поставщиков / покупателей**


Коды ТМЦ поставщиков / покупателей

Поставщик / покупатель	Код	Цена
0000 Закупка	F0000028084	280.60

Разрешить несколько кодов для одного поставщика |

**Просмотр изображения ТМЦ**

Основная карточка товара



Отобразить внутреннее изображение  
 Отобразить внешний файл

**Ввод PLU для товаров и весов**

Код	Артикул	Наименование	PLU
123456	121000-17	Основная карточка товара 1	8
642		Помидоры свеж.	12
2045		Помидоры некондиция	4

**Штрих коды ТМЦ**

ТМЦ | Основная карточка товара

4690501005039

**Допустимые комбинации характеристик ресурса \***

Основная карточка товара

Код	Артикул	Наименование	Штриховые коды
123456	121000-17	Основная карточка товара	4690501005039
123457	121000-17	Основная карточка товара 44 / Красный	

**Ввод единиц измерения ТМЦ**

Основная карточка товара

Связать коэффициенты

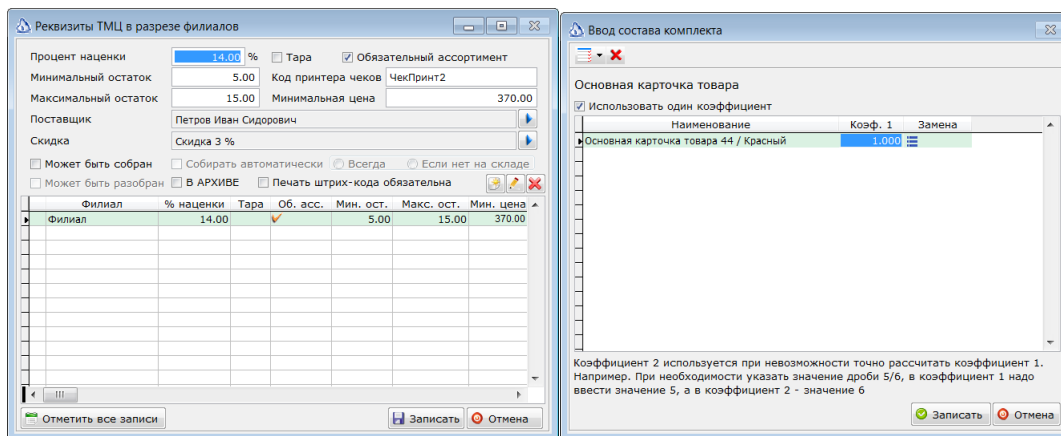
Единица	Количество	Цена	Штрих-код
упаковка из 10 шт	10.00000000	0.10000000	

**Ввод альтернативных товаров**

Основная карточка товара

Использовать один коэффициент

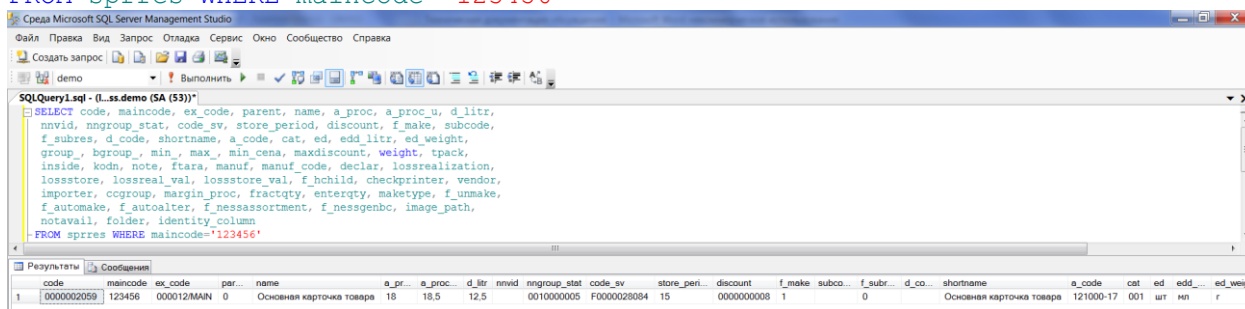
Наименование	Коэф. 1
Основная карточка товара 44 / Красный	1.000



Начнем рассмотрение с основной таблицы справочника товаров – `sprnn`. Следует помнить, что `sprnn` – это не таблица, а представление основанное на таблице `sprres`. Все используемые ниже запросы будут извлекать данные именно из этой таблицы.

Выполним запрос, который вернет нам основные реквизиты карточки товара. Идентифицировать товар будем ко Коду в главной базе. Если он указан в карточке, то он является уникальным. Данная проверка производится в триггере на добавление и обновление записей таблицы `sprres`.

```
SELECT code, maincode, ex_code, parent, name, a_proc, a_proc_u, d_litr,
nvid, nngroup_stat, code_sv, store_period, discount, f_make, subcode,
f_subres, d_code, shortname, a_code, cat, ed, edd_litr, ed_weight,
group, bgroup, min, max, min_cena, maxdiscount, weight, tpack,
inside, kodn, note, ftara, manuf, manuf_code, declar, lossrealization,
lossstore, lossreal_val, lossstore_val, f_hchild, checkprinter, vendor,
importer, ccgroup, margin_proc, fractqty, enterqty, maketype, f_unmake,
f_automake, f_autoalter, f_nessassortment, f_nessgenbc, image_path,
notavail, folder, identity_column
FROM sprres WHERE maincode='123456'
```



В результате выполнения запроса получим строку содержащую реквизиты карточки товара. Рассмотрим подробнее полученный результат. Следует обратить внимание, что в таблице приведен список полей, которые имеют отношение к товару.

Имя поля	Значение	Комментарий
code	0000002059	Внутренний код товара в системе. Этот код используется во всех ссылках на товар. Поле соответствует полю <code>nn</code> в представлении <code>sprnn</code> .
maincode	123456	Код в главной базе. Этот код чаще всего используется при обмене между филиалами, как уникальный идентификатор товара в сети и для обмена с оборудованием.
ex_code	000012/MAIN	Дополнительный код. Обычно служит для интеграции с внешними системами, например при выгрузках или загрузках данных в/из другие системы.
parent	0	Указатель на родительскую папку. Так. Справочник товаров используется в точке зрения и у него есть отдельная структура, это поле содержит 0.

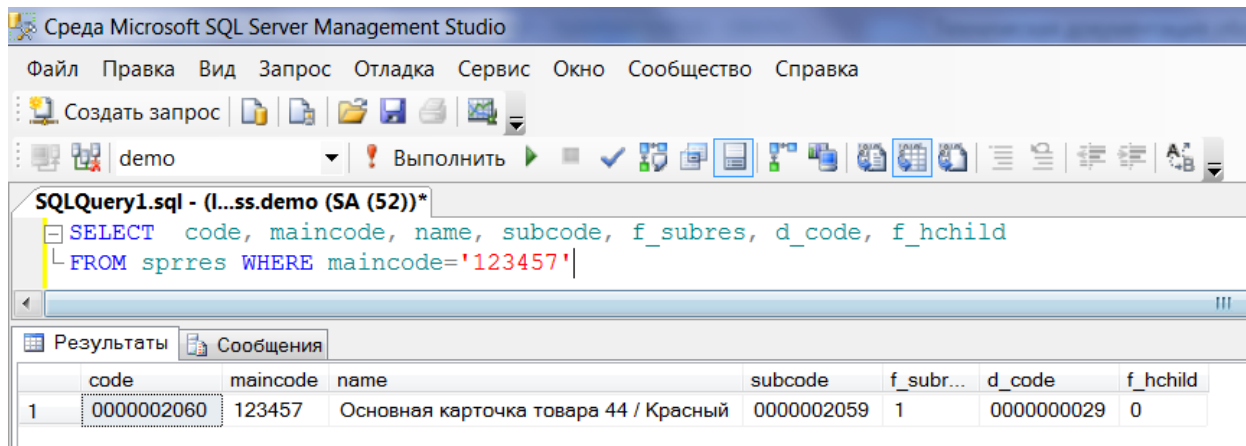
name	Основная карточка товара	Наименование товара.
a_proc	18	Поле крепость. Используется для учета алкогольной продукции.
a_proc_u	18.5	Поле крепость поставщика. Используется для учета алкогольной продукции. Отражает крепость продукции заявленную в документации от поставщика.
d_litr	12,5	Объем. Объем основной единицы товара.
nnvid		Вид алкогольной продукции. Используется при формировании различных отчетных документов.
nngroup_stat	0010000005	Группа статистики. Данное поле может использоваться для дополнительной группировки товара в различных отчетах.
code_sv	F0000028084	Поле Код поставщика.
store_period	15	Срок хранения товара в днях.
discount	0000000008	Код скидки на товар из справочника скидок. Исползуется для выгрузки в ККМ.
f_make	1	Признак, того что товар может быть собран из других товаров.
subcode		Код родительской карточки для подчиненного (характеристики) товара. В данном случае это основная карточка, поэтому код пустой.
f_subres	0	Признак подчиненного товара. В данном случае товар не является подчиненным (характеристикой) другому товару.
d_code		Код комбинации характеристик для подчиенного товара. В данном случае пустой.
shortname	Основная карточка товара	Краткое наименование товара.
a_code	121000-17	Артикул товара. Поле соответствует полю <code>code</code> в представлении <code>sprnn</code> .
cat	001	Категория. Для товаров всегда 001. Для услуг 003, для объектов ОС 003. Данное поле отсутствует в представлении <code>sprnn</code> .
ed	шт	Основная единица измерения товар.
edd_litr	мл	Единица измерения объема товара.
ed_weight	г	Единица измерения веса товара.
group_	0010000006	Основная группа товара.
bgroup_		Бухгалтерская группа товара.
min_	10	Минимальный запас.
max_	30	Максимальный запас.
min_cena	350	Минимальная цена продажи.
maxdiscount	15	Максимально допустимая скидка. Поле используется для выгрузки в ККМ.
weight	5	Вес основной единицы товара.
tpack	Пакет	Вид основной упаковки товара.
inside	20	Наиболее часто используемое вложение основных единиц в упаковку.
kodn	04	Код из справочника списков налогов ( <code>sprlistkodn</code> ).
note	Карточка создана для демонстрации хранения	Комментарий к карточке товара.

	реквизитов в базе данных	
ftara	0	Признак того, что чаще всего данная карточка является оборотной тарой.
manuf	0000010353	Код страны происхождения товара ( <i>sprcountry</i> ).
manuf_code	0000000007	Код производителя товара ( <i>sprclient</i> ).
declar	10126010/310810/0013370	Последний оприходованный номер грузовой таможенной декларации
lossrealization	0000000098	Код из справочника естественной убыли при реализации ( <i>sprloss</i> ).
lossstore	0000000154	Код из справочника естественной убыли при хранении ( <i>sprloss</i> ).
lossreal_val	5	Значения естественной убыли при реализации, указанные вручную.
lossstore_val	7	Значения естественной убыли при хранении, указанные вручную.
f_hchild	1	Признак наличия подчиненных карточек (характеристик). У данной карточки есть подчиненные.
checkprinter	ЧекПринт1	Код основного принтера чеков. Используется для выгрузки в ККМ.
vendor	0010076	Основной поставщик товара. Код из справочника контрагентов ( <i>sprclient</i> ).
importer		Основной импортер товара. Код из справочника контрагентов ( <i>sprclient</i> ). Используется для построения алкогольных деклараций.
ccgroup	0000000001	Код группы расчета цен.
margin_proc	15	Основной, рекомендуемый, процент наценки.
fractqty	0	Признак разрешения дробного количества. Используется для выгрузки в ККМ.
enterqty	0	Признак необходимости запрашивать количество. Используется для выгрузки в ККМ.
maketype	0	Способ сборки. 0 - Не раскладывать на комплектующие, 1 - Раскладывать на комплектующие всегда, 2 - Раскладывать на комплектующие, только если не хватает на складе
f_unmake	0	0 – товара не может быть разобран. Иначе 1.
f_automake	0	0 – товар не может быть автоматически собран. Иначе 1.
f_autoalter	0	0 – товар не может быть автоматически заменен на альтернативный. Иначе 1.
f_nessassortment	0	0 – товар не является обязательным ассортиментом. Иначе 1.
f_nessgenbc	0	Признак необходимости всегда генерировать штриховой код.
image_path		Путь и имя файла с изображением товара. В данном случае изображение загружено в базу, а не хранится в отдельном файле.
notavail	0	Признак того, что карточка перенесена в архив. В данном случае карточка не перенесена в архив.
folder	0	Признак того, что запись содержит информацию о папке товаров. В данном случае – это карточка товара, а не папка.
identity_column	8217	Уникальный идентификатор записи.

**Примечание.** Следует отметить, что в представлении *sprnn* следующие поля *min\_*, *max\_*, *min\_cena*, *ftara*, *vendor*, *checkprinter*, *margin\_proc*, *maketype*, *f\_make*, *f\_unmake*, *f\_automake*, *f\_nessassortment*, *f\_nessgenbc*, *discount*, *notavail* могут быть выбраны из таблицы *sprges\_own* (реквизиты в разрезе филиалов), поэтому их значения могут не совпадать со значениями в таблице *sprges*.

Теперь рассмотрим несколько полей у подчиненной карточки (характеристики) . Для этого изменим предыдущий запрос на следующий:

```
SELECT code, maincode, name, subcode, f_subres, d_code, f_hchild
FROM sprres WHERE maincode='123457'
```



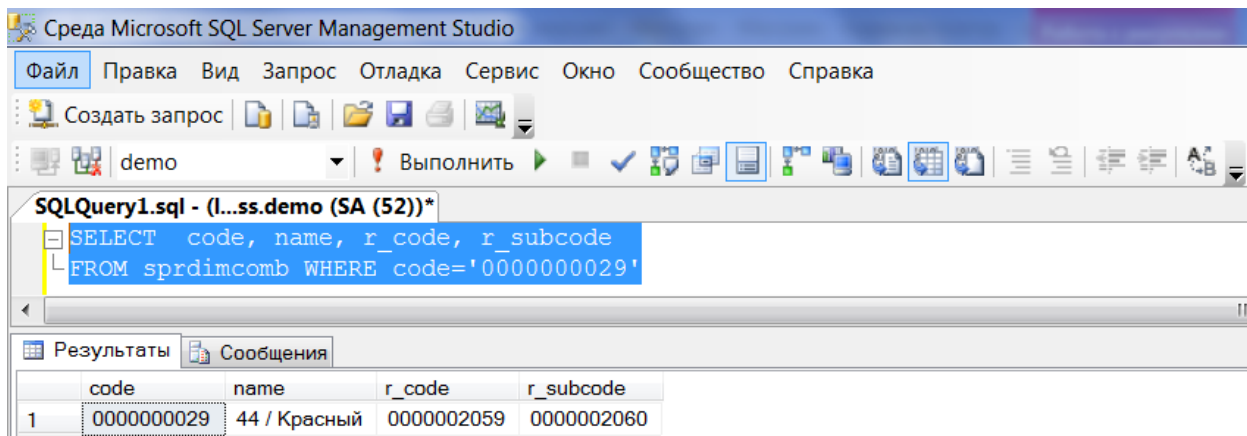
Имя поля	Значение	Комментарий
code	0000002060	Внутренний код товара в системе. Содержит уникальный новый код, т.к. характеристика является самостоятельной карточкой товара.
maincode	123457	Код в главной базе. Содержит новое уникальное значение.
name	Основная карточка товара 44 / Красный	Наименование подчиненной карточки, по умолчанию, генерируется на основании наименования основной карточки, с добавлением информации о выбранных характеристиках.
subcode	0000002059	Внутренний код в системе роительской карточки.
f_subres	1	Признак того, что карточка является подчиненной (характеристикой) к другой карточке.
d_code	0000000029	Код комбинации введенных характеристик из справочника комбинаций характеристик ( <a href="#">sprdimcomb</a> ).
f_child	0	Данная карточка не содержит других подчиненных карточек.

**Примечание.** Все остальные поля, по умолчанию, соответствуют полям родительской карточки. В последствии, любой реквизит подчиненной карточки может быть изменен самостоятельно на любое значение. Если требуется поддержание каких либо реквизитов подчиненных карточек (артикула, наименования, единицы измерения и т.п.) в соответствии со значениями основной карточки, то необходимо использовать скрипт проверки при записи основной карточки. Например, может быть такой скрипт:

```
// Обновление полей связанных карточек
// Обновляем название дочерних карточек
ЗАПРОС("UPDATE sprnn SET name= dbo.fn_getsubname( subcode, nn, '/' ), manuf= '' + manuf + '',
        ed= '' + ed + '', code= '' + code + ''
        WHERE subcode= '' + nn + ''");
```

Теперь рассмотрим как записаны характеристики в справочник комбинаций характеристик. Для этого выполним следующие запросы. Следует обратить внимание, что в качестве условия указан код комбинации из поля d\_code результата предыдущего запроса.

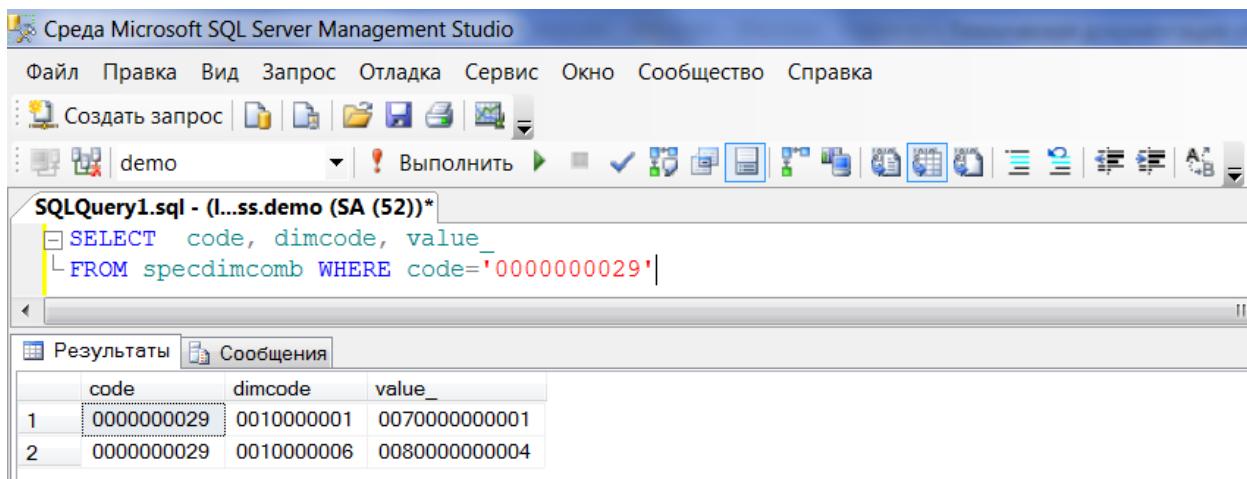
```
SELECT code, name, r_code, r_subcode
FROM sprdimcomb WHERE code='0000000029'
```



Имя поля	Значение	Комментарий
code	0000000029	Код комбинации характеристик. Соответствует значению поля d_code подчиненной карточки товара.
name	44 / Красный	Наименование комбинации. Оно формируется автоматически из названий выбранных характеристик.
r_code	0000002059	Код основной карточки товара. Соответствует полю subcode подчиненной карточки товара.
r_subcode	0000002060	Код подчиненной карточки товара. Соответствует полю pp подчиненной карточки товара.

Сами значения характеристик хранятся в многострочной части справочника комбинаций (sprdimcomb). Следующий запрос вернет нам в качестве результата список значений характеристик.

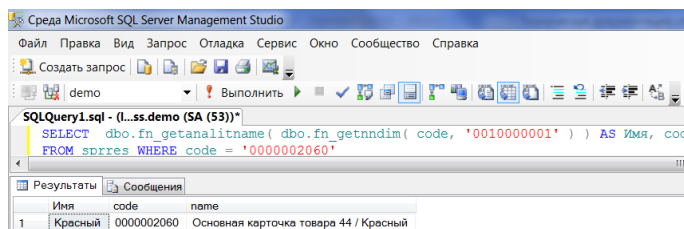
```
SELECT code, dimcode, value_
FROM sprdimcomb WHERE code='0000000029'
```



Имя поля	Значение	Комментарий
code	0000000029	Код комбинации характеристик.
dimcode	0010000001	Код характеристики из справочника характеристик (sprdim).
value_	0070000000001	Значение характеристики. Значение состоит из двух частей: <ol style="list-style-type: none"> <li>1. Код из справочника аналитик (spranalit). В данном случае 007.</li> <li>2. Код из справочника, на который ссылается указанная аналитика. В данном случае 0000000001.</li> </ol>

**Примечание.** В текущей реализации системы заблокировано добавление характеристик с типом отличным от Аналитика. Это связано с тем, что большинство внешних систем не поддерживают произвольные характеристики (разрезы), а только в виде справочников. Поэтому, все значения характеристик представляют собой значения состоящие из кода аналитики и кода из связанного справочника. Для определения наименования характеристики из значения в указанном формате необходимо воспользоваться хранимой функцией `fn_getanalitname( @analitvalue varchar(23) )`. Пример.

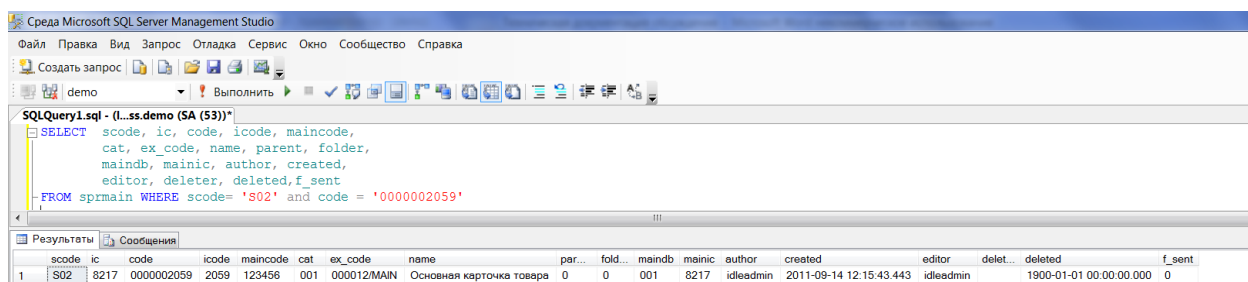
```
SELECT dbo.fn_getanalitname( dbo.fn_getnndim( code, '0010000001' ) ) AS Имя,
       code, name
FROM sprres WHERE code = '0000002060'
```



### Таблица sprmain.

Как было сказано выше, в момент добавления записи в основную таблицу справочника, автоматически добавляется запись в таблицу `sprmain`, содержащую общую информацию из всех справочников. Следующий запрос вернет нам в качестве результата запись, соответствующую Основной карточке товара.

```
SELECT      scode, ic, code, icode, maincode,
           cat, ex_code, name, parent, folder,
           maindb, mainic, author, created,
           editor, deleter, deleted, f_sent
FROM sprmain WHERE scode= 'S02' and code = '0000002059'
```



Имя поля	Значение	Комментарий
scode	S02	Код типа справочника. S02 – справочник товаров.
ic	8217	Идентификатор записи в основной таблице справочника. Соответствует полю identity_column
code	0000002059	Код карточки в справочнике.
icode	2059	Целочисленное представление кода. Используется для генерации новых кодов карточек.
maincode	123456	Соответствует полю maincode карточки товара.
cat	001	Соответствует полю cat записи в таблице sprres.
ex_code	000012/MAN	Соответствует полю ex_code карточки товара.
name	Основная карточка товара	Соответствует полю name карточки товара.
parent	0	Соответствует полю parent карточки товара.
folder	0	Соответствует полю folder карточки товара.

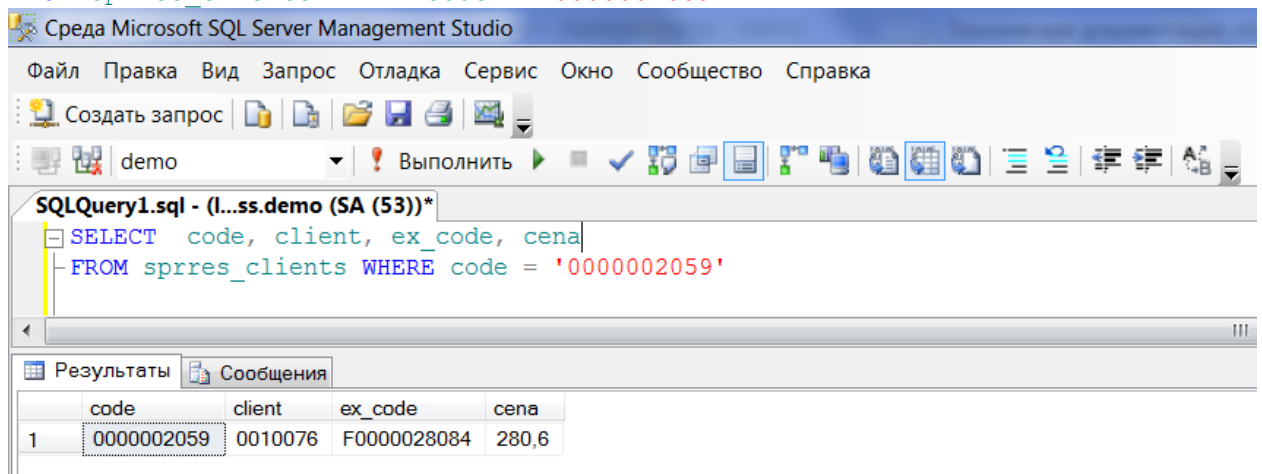


maindb	001	Код филиала, в котором была создана карточка.
mainic	8217	Идентификатор записи карточки в базе филиала, создавшего карточку. В данном случае соответствует полю ic, т.к. карточка была создана в текущем филиале.
author	idleadmin	Пользователь, создавший карточку. Заполняется автоматически при добавлении записи в таблицу <code>sprmain</code> .
created	2011-09-14 12:15:43.443	Дата и время создания карточки. Заполняется автоматически при добавлении записи в таблицу <code>sprmain</code> .
editor	idleadmin	Пользователь, последний корректировавший карточку товара.
deleter		Пользователь, удаливший карточку товара.
deleted	1900-01-01 00:00:00.000	Дата и время удаления карточки из базы данных. Указанное значение соответствует «пустой», неуказанной дате.
f_sent	0	Признак, указывающий, была ли карточка передана в другой филиал. В данном случае – не было передано.

Коды товара для поставщиков/покупателей (`sprres_clients`).

Коды поставщиков и покупателей хранятся в таблице `sprres_clients`. Для просмотра введенной в карточку товара информации выполним следующий запрос. В качестве условия укажем внутренний код Основной карточки товара.

```
SELECT code, client, ex_code, cena
FROM sprres_clients WHERE code = '0000002059'
```



Имя поля	Значение	Комментарий
code	0000002059	Код карточки товара в справочнике.
client	0010076	Код из справочника контрагентов, для которого введена информация.
ex_code	F0000028084	Указанный код, используемый контрагентом.
cena	280,6	Указанная цена, по прайс листу контрагента.

Изображения товара (`sprres_image`).

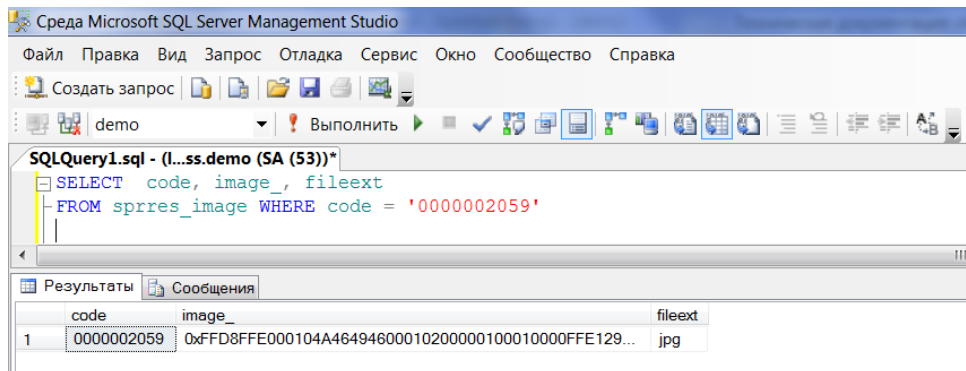
Система «Айтида» предлагает два способа указания изображения, соответствующего товару.

1. Загрузить изображение в таблицу `sprres_image`.

2. Указать путь и имя файла на внешнем носителе, в котором хранится изображение.

При втором варианте путь и имя файла хранятся в основной таблице справочника (см. выше). Для просмотра изображения загруженного в таблицу базы данных выполним следующий запрос.

```
SELECT code, image_, fileext
FROM sprres_image WHERE code = '0000002059'
```

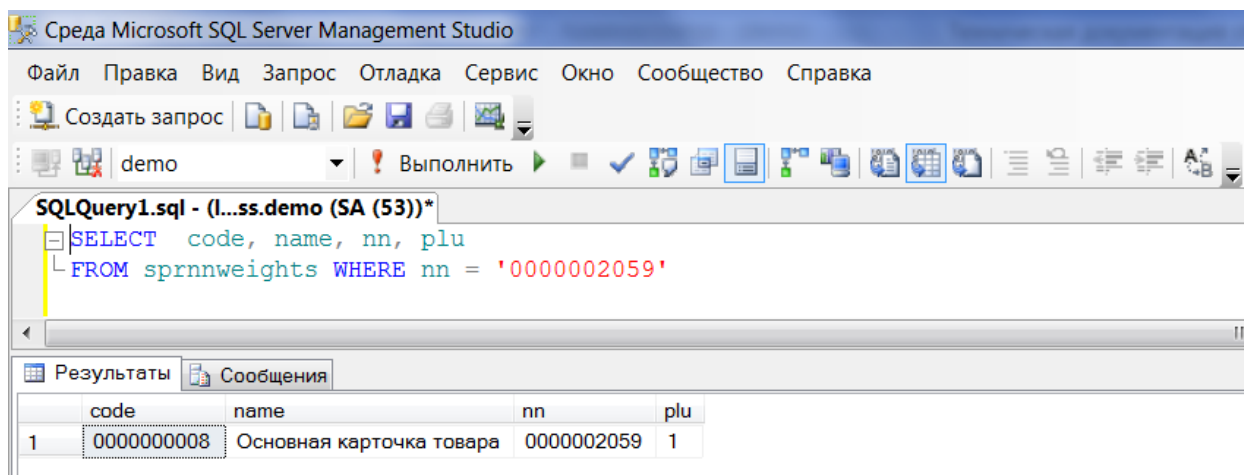


Имя поля	Значение	Комментарий
code	0000002059	Код карточки товара в справочнике.
image_	0xFFD8FFE000104A4649...	Содержимое загруженного файла в двоичном виде.
fileext	jpg	Тип (расширение имени) исходного файла.

Привязка товара к весам и PLU (sprnnweights).

Система «Айтида» предусматривает возможность выгрузки товаров одновременно в разные веса с печатью этикеток. При этом в каждом весе (группе весов) у товара может быть свой PLU. Для просмотра привязки товаров к весам выполним следующий запрос.

```
SELECT code, name, nn, plu
FROM sprnnweights WHERE nn = '0000002059'
```



Имя поля	Значение	Комментарий
code	0000000008	Код группы весов.
name	Основная карточка товара	Наименование товара для указанной группы весов.
nn	0000002059	Код карточки товара.

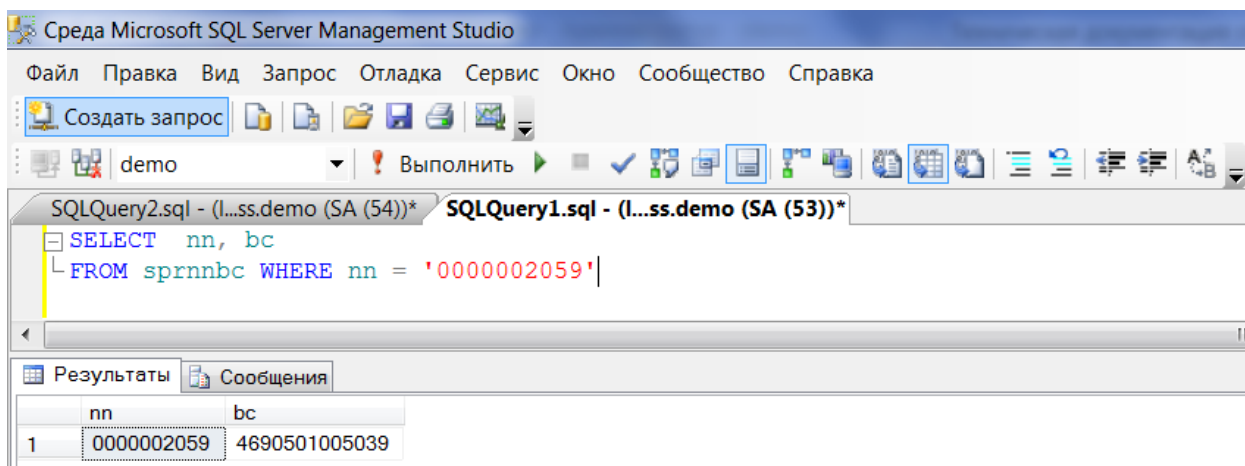
plu	1	Номер PLU товара для группы весов.
-----	---	------------------------------------

**Примечание.** Отдельного справочника, описывающего группы весов в системе нет. Коды групп генерируются динамически в процессе ввода новых групп в форму **Ввод PLU для товаров и весов**.

Штриховые коды товара (sprnbc).

Штриховые коды товаров, соответствующие основной единице измерения товара хранятся в таблице sprnbc. Используем следующий запрос для просмотра содержимого этой таблицы.

```
SELECT nn, bc
FROM sprnbc WHERE nn = '0000002059'
```



Имя поля	Значение	Комментарий
nn	0000002059	Внутренний код Основного товара.
bc	4690501005039	Присвоенный штриховой код.

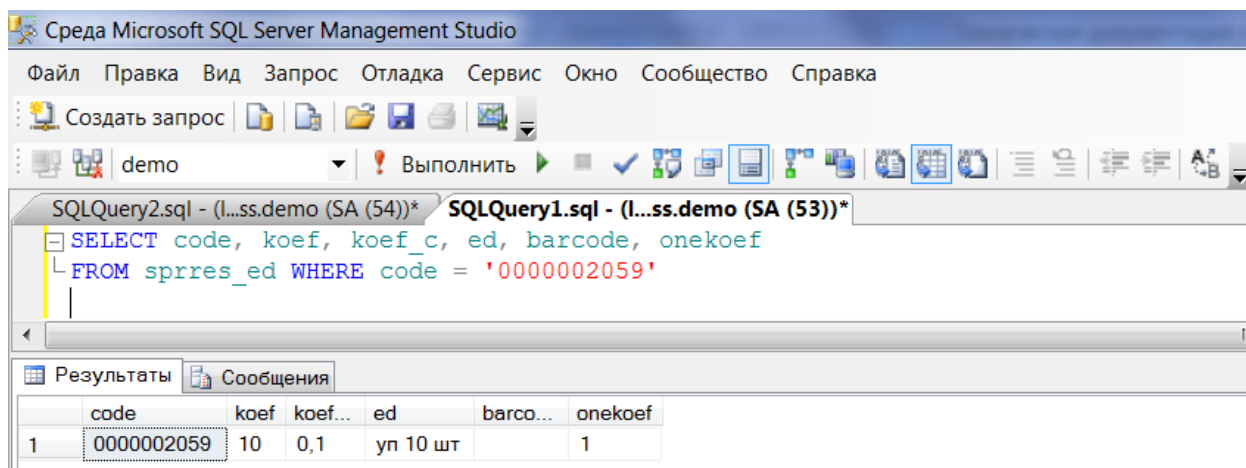
**Примечание.**

1. У одного товара может быть неограниченное количество штриховых кодов, определенных для основной единицы измерения.
2. При создании подчиненной (характеристики) карточки товара штриховые коды не копируются в новую карточку.

Дополнительные единицы измерения товара (sprres\_ed).

Во многих случаях помимо основной единицы измерения у товара могут быть дополнительные единицы измерения. Например, килограмм и грамм, штука и упаковка, пачка и блок и т.п. При этом, дополнительные единицы измерения могут иметь собственные штриховые коды и цена товара в дополнительной единице может быть другой. Например, штриховой код блока сигарет может иметь собственный штриховой код, отличный от штрихового кода отдельной пачки, а цена блока может быть меньше чем цена аналогичного количества пачек. Все эти настройки вводятся в окне дополнительных единиц измерения товара. Для просмотра того как указанная информация сохранена в базе данных выполним следующий запрос.

```
SELECT code, koef, koef_c, ed, barcode, onekoef
FROM sprres_ed WHERE code = '0000002059'
```



Имя поля	Значение	Комментарий
code	0000002059	Внутренний код Основного товара.
koef	10	Коэффициент пересчета количества товара из дополнительной единицы измерения в основную.
koef_c	0,1	Коэффициент пересчета цены товара из дополнительной единицы измерения в основную. Этот коэффициент может не быть равен 1/koef, что позволит отдельно рассчитывать цену товара для дополнительной единицы измерения.
ed	уп 10 шт	Код дополнительной единицы измерения.
barcode		Штриховой код, указывающий, что используется дополнительная единица измерения.
onekoef	1	Признак, если он равен 1, то это означает, что поле koef_c строго равно 1/koef и пользователю не требуется вводить поле koef_c вручную. Иначе, поле koef_c может принимать произвольное, отличное от 0 значение.

**Примечание.** При выборе в документе в строке товара дополнительной единицы измерения, система производит следующие расчеты нового количества и цены товара:

*КоличествоВДополнительнойЕдиницеИзмерения = КоличествоВОсновнойЕдиницеИзмерения / koef*

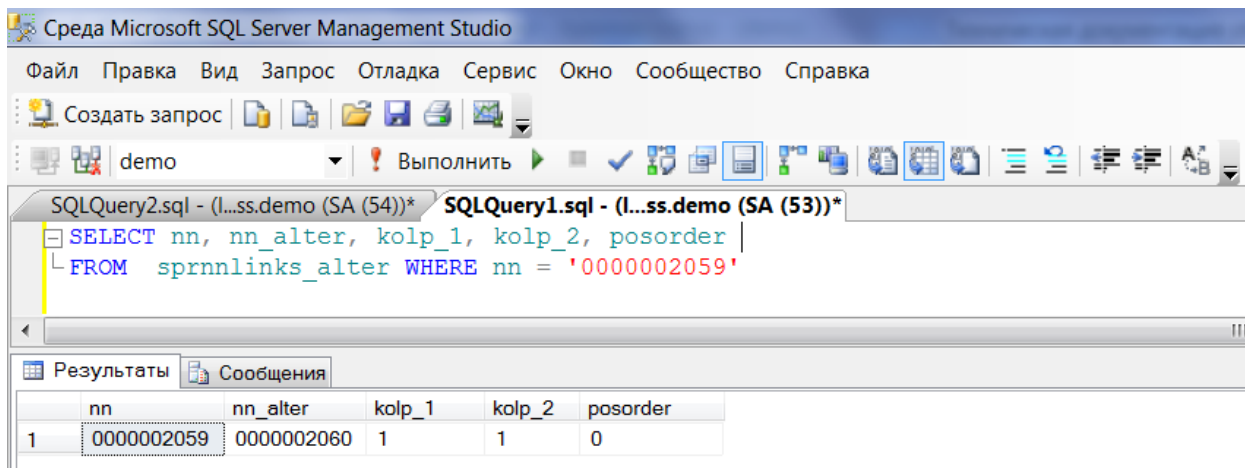
*ЦенаВДополнительнойЕдиницеИзмерения = ЦенаВОсновнойЕдиницеИзмерения / koef\_c*

В результате, если коэффициенты связаны между собой, т.е. koef \* koef\_c = 1, то сумма в строке документа не изменится. Если какой либо коэффициент ошибочно принял значение 0, то будет считаться, что он равен 1.

### Альтернативные товары (sprnnlinks\_alter).

Довольно часто, при производстве продукции возникают ситуации, при которых не хватает каких-либо ингредиентов. При этом, рецептура позволяет заменять недостающие ингредиенты на аналогичные, возможно в другом количестве. Система Айтида позволяет указывать и использовать альтернативные замены товаров на их аналоги. Для просмотра информации об альтернативных заменах выполним следующий запрос.

```
SELECT nn, nn_alter, kolp_1, kolp_2, posorder
FROM sprnnlinks_alter WHERE nn = '0000002059'
```



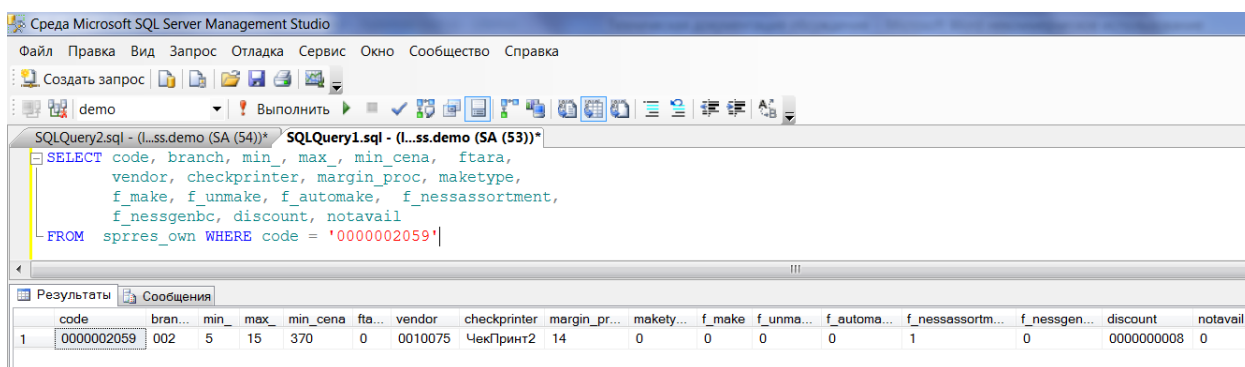
Имя поля	Значение	Комментарий
nn	0000002059	Внутренний код Основного товара.
nn_alter	0000002060	Внутренний код товара, на который можно заметить Основной товар.
kolp_1	1	Коэффициент (делимое) пересчета потребности в альтернативном ингредиенте от потребности в исходном ингредиенте.
kolp_2	1	Коэффициент (делитель) пересчета потребности в альтернативном ингредиенте от потребности в исходном ингредиенте.
posorder	0	Номер альтернативной замены. Замены будут производиться в указанном порядке.

**Примечание.** Итоговый коэффициент пересчета равен  $kolp_1 / kolp_2$ .

Реквизиты товара в разрезе филиалов (sprges\_own).

Для обеспечения управления реквизитами товара, значения которых могут отличаться в разных филиалах сети предусмотрен механизм ввода этих реквизитов в разрезе филиалов и передача необходимых данных средствами Айтиды в требуемые филиалы. Для просмотра того, как хранятся эти реквизиты, выполним следующий запрос.

```
SELECT code, branch, min_, max_, min_cena, ftara,
       vendor, checkprinter, margin_proc, maketype,
       f_make, f_unmake, f_automake, f_nessassortment,
       f_nessgenbc, discount, notavail
FROM sprges_own WHERE code = '0000002059'
```



Имя поля	Значение	Комментарий
code	0000002059	Внутренний код Основного товара.
branch	002	Код филиала, для которого определены реквизиты, и в который они будут переданы процедурами обмена данными.
min_	5	Минимальный запас в филиале

max_	15	Максимальный запас в филиале
min_cena	370	Минимальная цена продажи в филиале
ftara	0	Признак тары для товара в филиале
vendor	0010075	Основной поставщик товара в филиал
checkprinter	ЧекПринт2	Код принтера чеков в филиале
margin_proc	14	Рекомендуемый процент наценки в филиале.
maketype	0	Способ сборки товара в филиале
f_make	0	Товар может быть собран в филиале.
f_unmake	0	Товар может быть разобран в филиале.
f_automake	0	Товар может быть автоматически собран в филиале.
f_nessassortment	1	Товар входит в обязательный ассортимент филиала.
f_nessgenbc	0	Для товара необходимо всегда генерировать штриховые коды в филиале.
discount	0000000008	Основная скидка на товар в филиале.
notavail	0	Признак того, что карточка перенесена в архив в филиале.

**Примечание.** При передаче данных, содержащих данный товар в филиал, необходимо, чтобы шаблон обмена данными содержал правила синхронизации для справочника контрагентов и скидок клиентам.

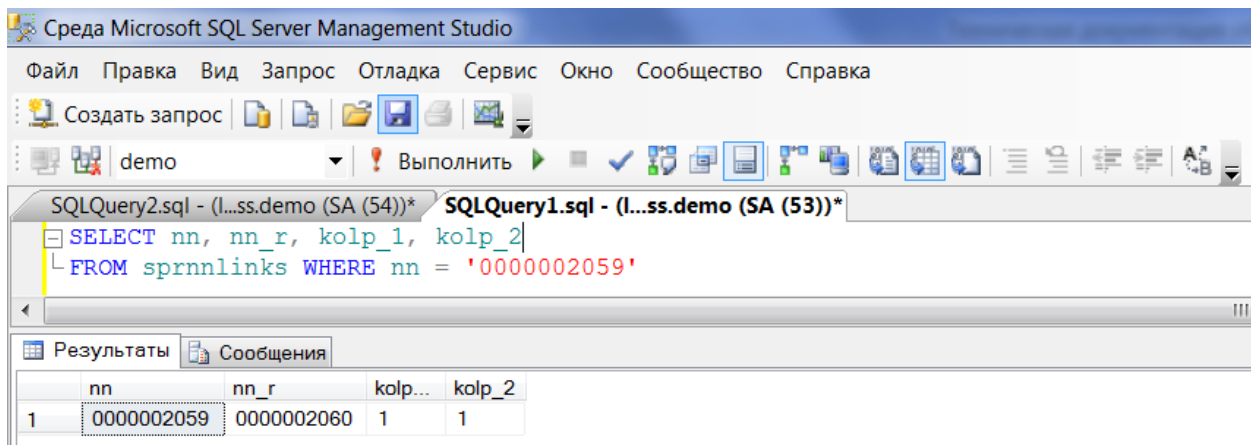
### Состав комплектов (sprnnlinks).

Состав для собираемых товаров в системе Айтида может быть задан двумя способами.

1. Документами Калькуляция, в которых указываются подробные правила приготовления блюда из ингредиентов. Для одного блюда в системе может быть определено несколько калькуляций, которые могут действовать в разные периоды времени.
2. Указанием состава изделия в карточке товара. Состав – это максимально упрощенная калькуляция, которая действует бессрочно во все периоды времени.

Для просмотра состава изделия выполним следующий запрос.

```
SELECT nn, nn_r, kolp_1, kolp_2
FROM sprnnlinks WHERE nn = '0000002059'
```



Имя поля	Значение	Комментарий
nn	0000002059	Внутренний код Основного товара.
nn_r	0000002060	Внутренний код товара, являющегося составной части изделия.
kolp_1	1	Коэффициент (делимое) расчета потребности в составной части при сборке изделия.
kolp_2	1	Коэффициент (делитель) расчета потребности в составной части при сборке изделия.

**Примечание.** Итоговый коэффициент расчета равен kolp\_1 / kolp\_2.

## Регистры учета.

Для сбора и анализа остатков и движений ресурсов предприятия, система использует регистры учета. Под ресурсами, в данном случае понимается все, что может быть использовано компанией для достижения своих целей, для удовлетворения собственных потребностей и потребностей внешних контрагентов. Т.е. к ресурсам относятся денежные средства, запасы товаров и материалов, дебиторская и кредиторская задолженности, основные средства и нематериальные активы, сотрудники и т.д.

Регистры учета являются механизмом конфигурации системы. Регистры автоматически изменяются документами при принятии документов к учету (проведении) так, как это предусмотрено в конфигурации. Данные регистров используются системой при построении различных отчетов о наличии и движении различных категорий ресурсов.

Запись информации о движении ресурсов в регистры может производиться документами на основании predetermined правил, заложенных в систему или на основании правил указанных при конфигурировании системы для нужд конкретного проекта. Каждая такая запись имеет определенное место на оси времени. Каждый документ может порождать неограниченное количество движений по различным регистрам.

### Основные регистры системы:

В следующей таблице приведен список регистров используемых в системе Айтида.

Код / номер регистра	Наименование	Имя таблицы	Комментарий
001	Регистр учета движений ТМЦ	reg001, reg_c001_balance, reg_c001	Регистр содержит все движения товаров в разрезе складов и МОЛ. Так же, в данном регистре рассчитывается средневзвешенная себестоимость в разрезе складов.
002	Регистр учета резервов ТМЦ	reg002	Регистр содержит информацию о резервировании товаров в разрезе складов и контрагентов.
003	Регистр учета движений денежных средств	reg003	Регистр содержит информацию о движении денежных средств в разрезе денежных карманов и валют.
004	Регистр учета расчетов с контрагентами	reg004	Регистр содержит информацию обо всех изменениях взаиморасчетов с контрагентами в валюте документов, информации о сопоставлениях документов и рассчитанных суммовых и курсовых разницах.
005	Регистр учета расчетов с филиалами	reg005	Регистр содержит информацию о движениях задолженностей между филиалами.
006	Регистр учета номеров партий ТМЦ	reg006	Регистр содержит все движения товаров в разрезе номеров партий и отвечает за реализацию схемы FIFO при списании товаров в разрезе партий.
007	Регистр учета залоговой тары	reg007	Регистр содержит информацию о движениях залоговой тары в разрезе номенклатуры товаров и контрагентов.
008	Регистр учета цен реализации	reg008	Регистр содержит все движения товаров в ценах реализации и все факты изменения цен реализации.
009	Регистр учета серийных номеров	reg009	Регистр содержит информацию о движении серийных номеров товаров.
010	Регистр учета проводок	reg010, reg_c010, reg_c010_bs, reg_c010_year, reg_c010_month	В данном регистре сохраняются проводки, которые создаются документами при выполнении бизнес – функций.
011	Регистр учета расчетов с сотрудниками	reg011	Регистр содержит информацию обо всех изменениях взаиморасчетов с сотрудниками собственных организаций.
012	Регистр учета заказов поставщикам	reg012	Регистр содержит информацию о заказах и их выполнении в разрезе товаров и поставщиков.
013	Регистр учета бизнес	reg013	В данном регистре сохраняются все движения бюджетных

	нот (показателей)		показателей, которые производятся документами при выполнении бизнес – функций.
014	Универсальный регистр учета	reg014	Универсальный регистр учета предназначен для обеспечения учета произвольных объектов, по правилам, настраиваемым в процессе конфигурирования системы.
015	Регистр учета вне оборотных активов	reg015	Регистр содержит всю информацию необходимую для учета объектов основных средств и прочих вне оборотных активов.
016	не используется	reg016	
017	Регистр учета ценных бумаг	reg017	Регистр содержит информацию о движении ценных бумаг в разрезе серий, номеров и эмитентов этих бумаг.
018	не используется	reg018	
019	Регистр учета комплектаций / калькуляций и проектов	reg019	Регистр содержит информацию о калькуляциях блюд, а также комплектации предназначенной для выполнения проектов по изготовлению объектов основных средств.

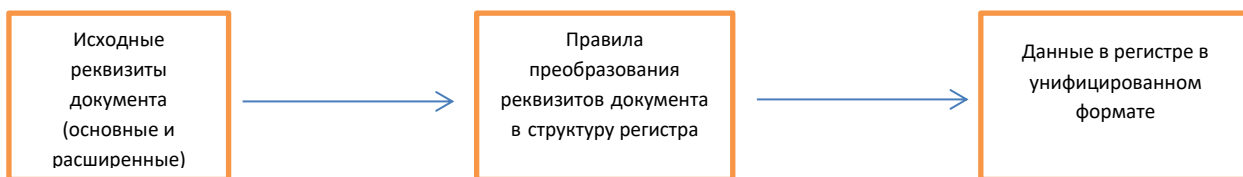
Далее рассмотрим наиболее важные аспекты, связанные с регистрами учета.

### Общие принципы хранения информации в регистрах.

Основным назначением регистров в системе является сбор информации из документов, которые осуществляют движения с объектом учета регистра и приведения этой информации к единому унифицированному виду. Это необходимо для того, чтобы получатели информации, учетные данные, отчеты и т.п., не зависели от появления новых или изменения функционала или структуры существующих документов. При добавлении или изменении документов достаточно добавить или изменить правила, по которым документ записывает данные в регистр. При этом сам регистр, его структура и смысл заполняющих его данных не изменится. Отсюда следует вывод, что все получатели информации должны брать информацию об объектах учета из регистров учета этих объектов.

Структура хранения информации в регистрах разработана таким образом, чтобы позволить пользователям, в случае необходимости, отменить проведение необходимых документов. При этом состояние системы и всех моделей учета должно вернуться в состояние, как если бы документ вообще не был проведен. С другой стороны, система имеет механизмы, позволяющие запретить выполнение подобных операций (отменять проведение).

Таким образом, в системе Айтида реализована следующая схема преобразования информации:



Правила преобразования реквизитов могут применяться как predefined, хранящиеся в служебной таблице transtable, так и пользовательские, заданные в процессе конфигурирования системы и хранящиеся в таблице tanstable\_ex.

### Общие реквизиты всех регистров.

Для реализации возможности отмены проведения документов, все регистры имеют следующие обязательные реквизиты, которые в обязательном порядке заполняются в момент проведения документа.

Имя поля	Тип данных	Назначение поля
code	char(3)	Тип документа создавшего запись.



ic	int	Идентификатор документа, создавшего запись.
ndok	char(20)	Номер документа, создавшего запись. Это информационное поле. Его значение никак не участвует в расчетах по регистру.
date	datetime	Дата проведения операции. Может отличаться от даты документа. Этой датой производятся все изменения и расчеты в регистре.
priz	char(1)	Признак прихода (+) или расхода (-) объекта учета. Если объект учета не подразумевает понятий приход/расход, например проводки по документу, то данный признак не используется.
account_	char(3)	Код модели учета, в которой произведено движение объекта учета.

Все остальные поля в регистрах уникальны и зависят от объекта учета регистра. Рассмотрим более подробно каждый регистр.

## Регистр учета движений ТМЦ.

Структура регистра.

Регистр учета ТМЦ состоит из трех таблиц.

1. Таблица *reg001* содержит все движения ТМЦ в разрезе складов и МОЛ. В эту таблицу документы помещают свою информацию о движениях ТМЦ.
2. Таблица *reg\_c001\_balance* содержит агрегированные данные о движении ТМЦ по дням в разрезе складов и МОЛ. В этой таблице производится расчет средневзвешенной учетной цены ТМЦ. Формула расчета будет указана ниже. Актуальность данных в таблице поддерживается автоматически триггерами системы.
3. Таблица *reg\_c001* содержит информацию о наличии и средневзвешенной себестоимости ТМЦ в разрезе складов и МОЛ на момент последнего зафиксированного движения ТМЦ (это может быть как текущий момент, так и будущее время). Таблица используется для быстрого доступа к текущим остаткам ТМЦ. Актуальность данных в таблице поддерживается автоматически триггерами системы.

Рассмотрим более подробно каждую из этих таблиц.

Таблица *reg001*.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-')
account_	char(3)	Код модели учета.
nn	char(10)	Внутренний код ТМЦ
sklad	char(3)	Код склада хранения ТМЦ.
mol	char(7)	Код МОЛ.
cena	float	Цена операции
kolp	float	Количество в операции
amount	float	Сумма операции
type	char(15)	Тип прихода/расхода (Закупка, бонус, нал/безнал и т.п.)
code_s	char(3)	Код документа резерва
ic_s	int	ИД документа резерва
firm	char(7)	Код организации. Информационное поле. В расчетах не участвует.
client	char(7)	Код контрагента. Информационное поле. В расчетах не участвует.
project	char(10)	Код проекта. Информационное поле. В расчетах не участвует.
transtime	datetime	Дата и время создания записи
_amount	float	не используется, всегда 0.
_kolp	float	не используется, всегда 0.
kolp_	float	не используется, всегда 0.
amount_	float	не используется, всегда 0.
_cena	float	не используется, всегда 0.
cena_	float	не используется, всегда 0.

### Примечание.

1. В момент добавления и удаления записей триггеры таблицы *reg001* производят необходимые изменения в таблицах *reg\_c001\_balance* и *reg\_c001*, для поддержки актуальности хранимых в этих таблицах данных. Все необходимые проверки – наличие на складе, “промежуточные минусы”, открытость отчетного периода, проверка наличия более поздних инвентаризаций, также производятся триггерами этой таблицы. Поэтому допускается удаление и добавление записей, непосредственно в таблицу регистра. Не допускаются изменения записей регистра командой UPDATE.
2. Если в карточке склада не указано, что необходимо вести учет ТМЦ в разрезе МОЛ, то система принудительно очищает поле *mol* в регистре учета ТМЦ. Поэтому, если в течение некоторого времени велся учет ТМЦ не в

- разреze МОЛ, то для ведения учета в разреze МОЛ необходимо провести инвентаризацию по каждому МОЛ, или оформить внутренние перемещения с “пустого” МОЛ на указанного. Аналогично, если велся учет в разреze МОЛ, то для перехода на ведение учета без МОЛ, необходимо провести либо инвентаризацию, либо оформить соответствующие внутренние перемещения. При оформлении указанных операций следует учитывать, что при ведении учета в разреze МОЛ и не указании МОЛ в документе, система автоматически установит значение МОЛ равным главному МОЛ склада, если такой определен.
3. Если в справочнике складов не указаны признаки, разрешающие списание “в минус”, то система проверяет текущий остаток на складе за вычетом активного резерва товара на этом же складе.
  4. При возникновении ошибок согласованности данных между таблицами `reg001`, `reg_c001_balance` и `reg_c001`, необходимо воспользоваться процедурой `sp_recalcreg001` для восстановления данных.
  5. В стандартной конфигурации документы при проведении влияют на регистр учета ТМЦ, только если склад, на котором осуществляется движение, находится в текущем филиале. В противном случае, документы создают движения в регистре учета расчетов с филиалами (`reg005`).
  6. В стандартной конфигурации документ внутреннее перемещение приходит ТМЦ на склад, только если установлен признак Приход и указана дата и время прихода. В зависимости от установленного флага “сторнировать расход”, документ внутреннее перемещение либо приходит ТМЦ на склад получатель (признак не установлен), либо списывает отрицательное количество со склада получателя, тем самым добавляя количество ТМЦ на склад получатель, без изменения учетной себестоимости.

### Таблица `reg_c001_balance`.

Основное назначение таблицы `reg_c001_balance` состоит в ускорении расчетов остатков и учетной себестоимости ТМЦ на любую дату и на любом складе. В этой таблице все данные по товару, складу и МОЛ суммируются в течение одного дня. Поэтому, при большом обороте номенклатуры выигрыш во времени и затраченных на расчеты ресурсах получается значительный.

Имя поля	Тип данных	Назначение поля
<code>date</code>	<code>datetime</code>	Дата, на которую производится расчет.
<code>nn</code>	<code>char(10)</code>	Код учитываемого ТМЦ
<code>sklad</code>	<code>char(3)</code>	Склад, на котором производится расчет остатка и себестоимости ТМЦ.
<code>mol</code>	<code>char(7)</code>	МОЛ на котором производится расчет остатка ТМЦ.
<code>account_</code>	<code>char(3)</code>	Код модели учета, для которой производится расчет.
<code>_cena</code>	<code>float</code>	Учетная себестоимость на складе указанного ТМЦ цена на начало дня.
<code>cena_</code>	<code>float</code>	Учетная себестоимость на складе указанного ТМЦ цена на конец дня.
<code>kolp_in</code>	<code>float</code>	Количество прихода указанного ТМЦ на указанный склад за день.
<code>kolp_out</code>	<code>float</code>	Количество расхода указанного ТМЦ с указанного склада за день.
<code>amount_in</code>	<code>float</code>	Общая сумма прихода ТМЦ на склад за день
<code>_kolp</code>	<code>float</code>	Остаток ТМЦ на складе на начало дня.
<code>kolp_</code>	<code>float</code>	Остаток ТМЦ на складе на конец дня.

**Примечание.** В момент добавления (удаления) записи в таблицу `reg001` система автоматически производит перерасчет в таблице `reg_c001_balance` за дату добавляемой (удаляемой) операции и за все последующие даты. Таким образом, производится автоматический пересчет всех учетных себестоимостей, даже если произведены изменения задним числом. Формула расчета учетной себестоимости выглядит следующим образом:

*РасчетноеКоличество = (ЕСЛИ (ОстатокНаНачалоДня < 0) ТО 0 ИНАЧЕ ОстатокНаНачалоДня ) + КоличествоПриходаЗаДень;  
 РасчетнаяСумма = (ЕСЛИ (ОстатокНаНачалоДня < 0) ТО 0 ИНАЧЕ ОстатокНаНачалоДня \* УчетнаяСебестоимостьНаНачалоДня ) + СуммаПриходаЗаДень;*

*УчетнаяСебестоимостьНаКонецДня = ЕСЛИ (РасчетноеКоличество == 0) ТО УчетнаяСебестоимостьНаНачалоДня ИНАЧЕ (РасчетнаяСумма / РасчетноеКоличество);*

Для расчета остатка ТМЦ на любую дату и время на любом складе необходимо использовать хранимую функцию:

`fn_calcsklad (@nn char(10), @date datetime, @sklad char(3), @account_ char(3), @mol char(7))`

Для расчета остатка ТМЦ на любую дату на начало дня на любом складе необходимо использовать хранимую функцию:

`fn_calcsklad_daybegin (@nn char(10), @date datetime, @sklad char(3), @account_ char(3), @mol char(7))`

Для расчета остатка ТМЦ на любую дату на конец дня на любом складе необходимо использовать хранимую функцию:

`fn_calcsklad_dayend (@nn char(10), @date datetime, @sklad char(3), @account_ char(3), @mol char(7))`

Для расчета себестоимости товара на указанную дату на любом складе необходимо использовать хранимую функцию:

`fn_calcsklad_incena (@nn char(10), @date datetime, @sklad char(3), @account_ char(3))`

Для расчета остатка ТМЦ с учетом всех подчиненных товаров необходимо использовать хранимую функцию:

`fn_calcskladsubres (@nn char(10), @date datetime, @sklad char(3), @account_ char(3), @mol char(7))`

Получение списка товаров с остатками на даты начала и окончания периода и оборотами за период:  
`sp_getnnsaldo_turnover( @nn char(10), @date_min datetime, @date_max datetime, @sklad char(3), @mol char(7), @account_char(3));`

### Таблица reg\_c001.

Основное назначение таблицы `reg_c001` состоит в ускорении расчетов остатков и учетной себестоимости товара на складах на текущий момент времени. Под текущим моментом понимается дата и время последней проведенной операции с указанным товаром.

Имя поля	Тип данных	Назначение поля
nn	char(10)	Код учитываемого ТМЦ
sklad	char(3)	Склад, на котором производится расчет остатка и себестоимости ТМЦ.
mol	char(7)	МОЛ на котором производится расчет остатка ТМЦ.
account_	char(3)	Код модели учета, для которой производится расчет.
cena	float	Учетная себестоимость на складе указанного ТМЦ.
kolp	float	Остаток указанного ТМЦ на складе и МОЛ.

**Примечание.** В момент добавления (удаления) записи в таблицу `reg001` система автоматически производит перерасчет в таблице `reg_c001` за последнюю дату, которая есть в регистре для ТМЦ, склада и МОЛ. Для расчета текущих остатка, себестоимости и остатка с учетом подчиненных ТМЦ необходимо использовать хранимые функции:

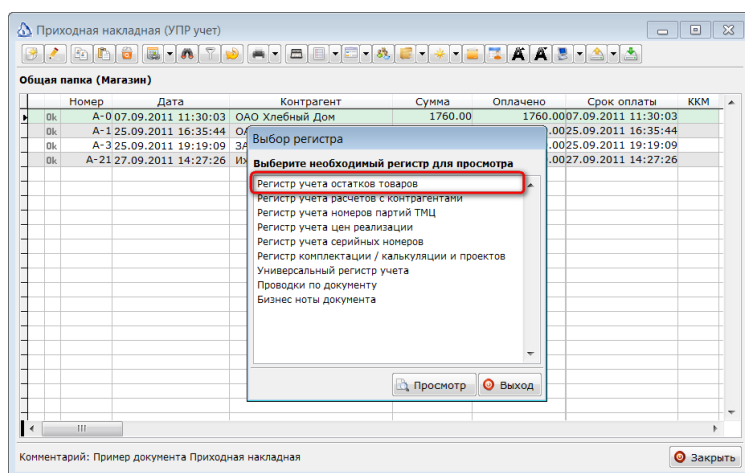
`fn_calccurrentsklad( @nn char(10), @sklad char(3), @account_char(3), @mol char(7))`

`fn_calccurrentsklad_incena( @nn char(10), @date datetime, @sklad char(3), @account_char(3))`

`fn_calccurrentskladsubres( @nn char(10), @date datetime, @sklad char(3), @account_char(3), @mol char(7))`

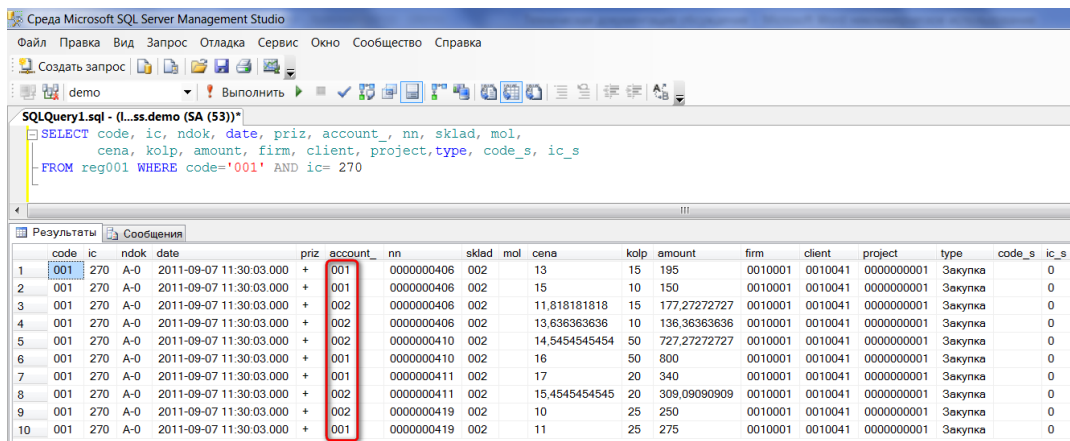
### Примеры заполнения регистра учета ТМЦ и использования функций.

В качестве исходного документа рассмотрим приходную накладную № А-0, которую мы ввели ранее. Проведем этот документ и посмотрим, на какие регистры он влияет.



Выполним запрос для просмотра содержимого регистра `reg001`.

```
SELECT code, ic, ndok, date, priz, nn, sklad, mol,
       account_, cena, kolp, firm, client, project,
       amount, type, code_s, ic_s
FROM reg001 WHERE code='001' AND ic= 270
```



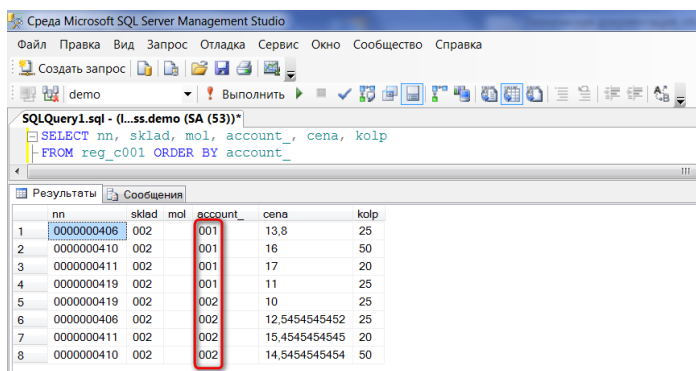
Как видно из рисунка в регистре записей в два раза больше чем строк в документе. Это потому что созданы записи по двум учетам – управленческому и бухгалтерскому. Обратите внимание, что по бухгалтерскому учету (код учета 002) цены и суммы указаны без НДС. Это происходит по тому, что в карточке Бухгалтерский учет в справочнике окружений установлен признак *'Вести учет ТМЦ без НДС'*. Рассмотрим подробнее содержимое первой записи.

Имя поля	Значение	Комментарий
code	001	Код приходной накладной
ic	270	Идентификатор нашего документа.
ndok	A-0	Номер документа
date	2011-09-07 11:30:03.000	Дата операции движения. В данном случае – дата документа.
priz	+	Признак прихода. Приходная накладная добавляет ТМЦ на склад.
account_	001	Код модели управленческого учета.
nn	0000000406	Внутренний код ТМЦ.
sklad	002	Код склада хранения ТМЦ.
mol		Код МОЛ. В документе не был указан МОЛ.
cena	13	Цена операции
kolp	15	Количество в операции
amount	195	Сумма операции
firm	0010001	Код организации.
client	0010041	Код контрагента.
project	0000000001	Код проекта.
type	Закупка	Тип прихода
code_s		Код документа резерва. Документ не был выписан на основании заказа поставщику или счета входящего ТМЦ.
ic_s	0	ИД документа резерва.

**Примечание.** В стандартной конфигурации, для всех документов, кроме Внутреннего перемещения и Пересортицы ТМЦ, записи создаются с датой, равной дате документа. Документы Внутреннего перемещения и Пересортицы ТМЦ создаются записи о списании ТМЦ с датой документа, а записи по приходу ТМЦ с датой прихода.

Выполним запрос для просмотра содержимого регистра *reg\_c001*. Так как рассматриваемый документ пока что единственный проведенный по регистру учета ТМЦ, никаких дополнительных условий на выборку данных не указываем.

```
SELECT nn, sklad, mol, account_, cena, kolp
FROM reg_c001 ORDER BY account_
```

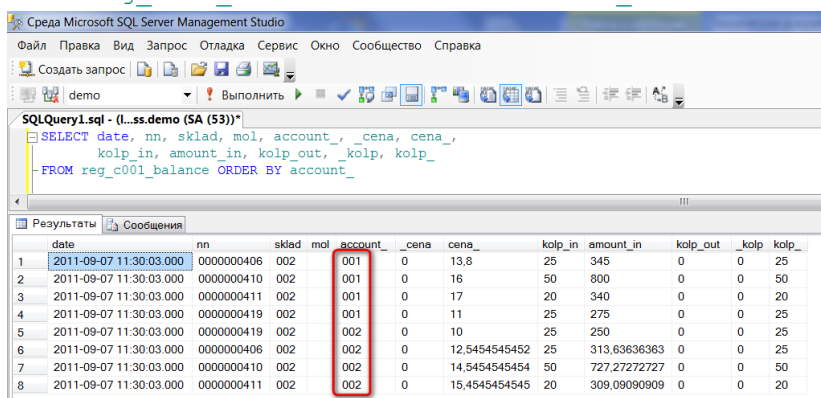


Обратите внимание, что в результате выборки данных отобразилось 8 строк. Четыре из них по управленческому учету и 4 по бухгалтерскому. Их 8, а не 10, как в таблице *reg001*, потому что в документе две строки содержали одинаковый товар с кодом 0000000406. Рассмотрим значения в первой строке выборки.

Имя поля	Значение	Комментарий
nn	0000000406	Код ТМЦ.
sklad	002	Код склада, на который мы оприходовали ТМЦ
mol		МОЛ не был указан.
account_	001	Код модели управленческого учета.
cena	13.8	Средневзвешенная себестоимость ТМЦ на указанном складе. Обратите внимание, что в исходном документе было две строки с данным ТМЦ по цене 13 и 15 и в количестве 15 и 10 соответственно. В результате средневзвешенного усреднения по формуле $(13 * 15 + 15 * 10) / 25 =$ получили значение 13.8.
kolp	25	Текущее количество ТМЦ на складе.

Выполним запрос для просмотра содержимого таблицы *reg\_c001\_balance*. Так как рассматриваемый документ пока что единственный проведенный по регистру учета ТМЦ, никаких дополнительных условий на выборку данных не указываем.

```
SELECT date, nn, sklad, mol, account_, _цена, цена_,
       kolp_in, amount_in, kolp_out, _kolp, kolp_
FROM reg_c001_balance ORDER BY account_
```



Имя поля	Значение	Комментарий
date	2011-09-07 11:30:03.000	Расчетная дата. Время у даты не учитывается при расчетах.
nn	0000000406	Код ТМЦ, для которого производится расчет себестоимости и оборота за день.
sklad	002	Код склада, для которого производится расчет себестоимости и оборота.
mol		Код МОЛ, для которого рассчитываются обороты.
account_	001	Код модели управленческого учета.

_cena	0	Средневзвешенная себестоимость на начало дня.
cena_	13.8	Средневзвешенная себестоимость на конец дня.
kolp_in	25	Количество поступлений ТМЦ за день.
amount_in	345	Сумма поступлений ТМЦ за день
kolp_out	0	Количество списаний (выбытий со склада) ТМЦ за день.
_kolp	0	Остаток ТМЦ на начало дня.
kolp_	25	Остаток ТМЦ на конец дня.

**Примечание.**

1. Уникальным ключом в таблице является список полей: *date (без времени), nn, sklad, mol, account\_*.
2. Поля *kolp\_in, kolp\_out, amount\_in* являются итоговыми суммами из таблицы *reg001* за указанный день, по указанному ТМЦ, складу и МОЛ.
3. Остаток ТМЦ на начало дня (поле *\_kolp*) равен остатку ТМЦ на конец предыдущего дня. Заполняется по этой же таблице из поля *kolp\_* за предыдущий день.
4. Остаток на конец дня (поле *kolp\_*) рассчитывается по формуле:  

$$kolp\_ = \_kolp + kolp\_in - kolp\_out$$
5. Средневзвешенная себестоимость на начало дня (поле *\_cena*) равна средневзвешенной себестоимости на конец предыдущего дня. Заполняется по этой же таблице из поля *cena\_* за предыдущий день.
6. Средневзвешенная себестоимость на конец дня (поле *cena\_*) рассчитывается по формуле (в формуле используются временные переменные, для упрощения записи):  

$$\begin{aligned} @kolp\_calc &= CASE WHEN \_kolp < 0 THEN 0 ELSE \_kolp END + kolp\_in \\ @amount\_calc &= CASE WHEN \_kolp < 0 THEN 0 ELSE \_kolp * \_cena END + amount\_in \\ cena\_ &= CASE @kolp\_calc = 0 THEN \_cena ELSE @amount\_calc / @kolp\_calc \end{aligned}$$

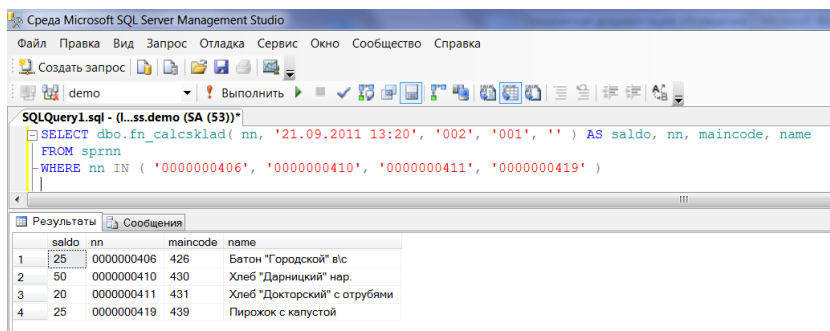
Пример использования функций для выборки данных.

*Функция fn\_calcsklad – расчет остатка ТМЦ на складе.*

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток.
@date	datetime	Дата и время, на которые необходимо произвести расчет.
@sklad	char(3)	Код склада, на котором необходимо определить остаток.
@account_	char(3)	Код модели учета для которой производится расчет.
@mol	char(7)	Код МОЛ, на котором необходимо определить остаток.

Выполним следующий запрос, в качестве условия укажем список ТМЦ, который был оприходован в примере:

```
SELECT dbo.fn_calcsklad( nn, '21.09.2011 13:20', '002', '001', '' ) AS saldo,
      nn, maincode, name
FROM sprnn
WHERE nn IN ( '0000000406', '0000000410', '0000000411', '0000000419' )
```



Первый столбец результата содержит запрошенное количество.

*Функция fn\_calcsklad\_daybegin – расчет остатка ТМЦ на складе на начало дня.*

*Функция fn\_calcsklad\_dayend – расчет остатка ТМЦ на складе на конец дня.*

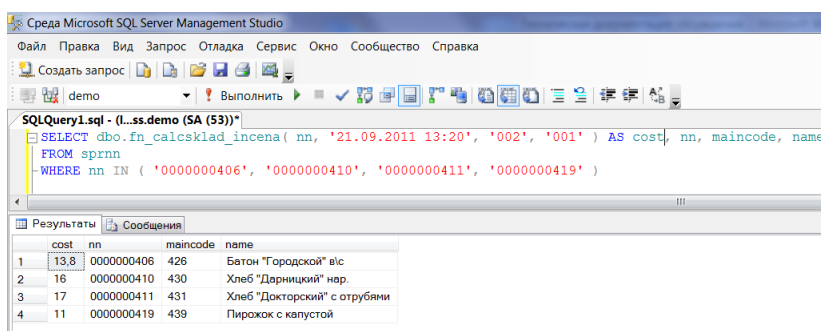
Эти функции полностью аналогичны *fn\_calcsklad*, только в указанной дате игнорируется время и расчет производится на начало или конец дня.

Функция *fn\_calcksklad\_incena* – расчет себестоимости ТМЦ на конец дня указанной даты на указанном складе.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток.
@date	datetime	Дата и время, на которые необходимо произвести расчет.
@sklad	char(3)	Код склада, на котором необходимо определить остаток.
@account_	char(3)	Код модели учета для которой производится расчет.

Выполним следующий запрос, в качестве условия укажем список ТМЦ, который был оприходован в примере:

```
SELECT dbo.fn_calcksklad_incena( nn, '21.09.2011 13:20', '002', '001' ) AS cost,
      nn, maincode, name
FROM sprnn
WHERE nn IN ( '0000000406', '0000000410', '0000000411', '0000000419' )
```



В первом столбце результата отображена средневзвешенная себестоимость ТМЦ на складе на указанную дату.

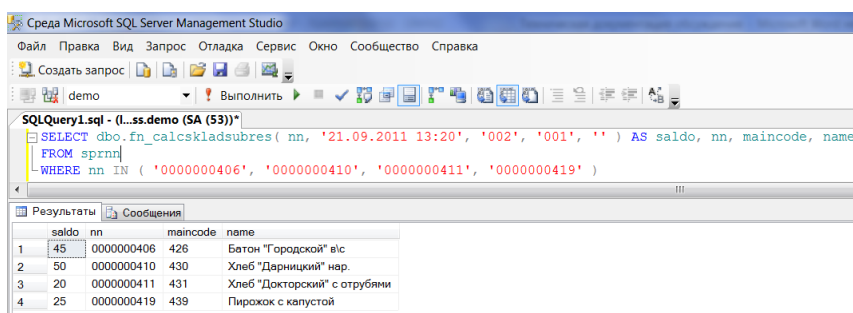
Функция *fn\_calckskladsubres* расчет количества остатка ТМЦ с учетом всех подчиненных ТМЦ.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток.
@date	datetime	Дата и время, на которые необходимо произвести расчет.
@sklad	char(3)	Код склада, на котором необходимо определить остаток.
@account_	char(3)	Код модели учета для которой производится расчет.
@mol	char(7)	Код МОЛ, на котором необходимо определить остаток.

Выполним следующий запрос, предварительно добавив в подчиненные к ТМЦ с кодом 0000000406 ТМЦ с кодом 0000000411.

```
UPDATE sprres SET subcode='0000000406', f_subres= 1 WHERE code= '0000000411'
```

```
SELECT dbo.fn_calckskladsubres( nn, '21.09.2011 13:20', '002', '001', '' ) AS
saldo, nn, maincode, name
FROM sprnn
WHERE nn IN ( '0000000406', '0000000410', '0000000411', '0000000419' )
```





Видим, что для ТМЦ с кодом 0000000406 рассчиталось количество с учетом подчиненной карточки с кодом 0000000411.

*Функции fn\_calcurrentsklad, fn\_calcurrentsklad\_incena, fn\_calcurrentskladsubres*

Эти функции полностью аналогичны вышеописанным, но расчет производится на с учетом всех проведенных в системе документов, а не на конкретную дату.

*Процедура sp\_getnnsaldo\_turnover расчет остатков и оборотов ТМЦ на складе за период.*

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток. Если не указан, то предполагается, что список кодов находится во временной таблице #getnnsaldo_turnover, у которой должно быть поле code, содержащее коды требуемых ТМЦ.
@date_min	datetime	Дата и время начала расчетного периода.
@date_max	datetime	Дата и время окончания расчетного периода.
@sklad	char(3)	Код склада, для которого необходимо произвести расчет.
@mol	char(7)	Код МОЛ, для которого необходимо произвести расчет.
@account_	char(3)	Код модели учета, для которой производится расчет.

Выполним эту процедуру, предварительно создав временную таблицу с четырьмя кодами ТМЦ.

```
SELECT code INTO #getnnsaldo_turnover
FROM sprres
WHERE code IN ( '0000000406', '0000000410', '0000000411', '0000000419' )

EXECUTE sp_getnnsaldo_turnover '', '01.09.2011', '23.09.2011', '002', '', '001'
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The SQL query window contains the following code:

```
SELECT code INTO #getnnsaldo_turnover FROM sprres WHERE code IN ( '0000000406', '0000000410', '0000000411', '0000000419' )
EXECUTE sp_getnnsaldo_turnover '', '01.09.2011', '23.09.2011', '002', '', '001'
```

The Results window displays the following data:

code	quantity_initial	quantity_closing	amount_initial	amount_closing	quantity_income	quantity_outcome	amount_income	amount_outcome
1 0000000406	0	25	0	375	25	0	375	0
2 0000000410	0	50	0	900	50	0	900	0
3 0000000411	0	20	0	380	20	0	380	0
4 0000000419	0	25	0	300	25	0	300	0

Имя поля	Тип значения	Комментарий
code	char(10)	Код ТМЦ
quantity_initial	float	Количество ТМЦ на начало периода.
quantity_closing	float	Количество ТМЦ на конец периода.
amount_initial	float	Сумма остатка ТМЦ на начало периода.
amount_closing	float	Сумма остатка ТМЦ на конец периода.
quantity_income	float	Количество ТМЦ поступившее за период.
quantity_outcome	float	Количество ТМЦ израсходованное за период.
amount_income	float	Сумма ТМЦ поступившая за период.
amount_outcome	float	Сумма ТМЦ израсходованная за период.

**Примечание.** Расчет количества производится для указанного склада и, если указан МОЛ, для указанного МОЛ. Расчет сумм производится в ценах учета склада.

## Регистр учета резервов ТМЦ.

Система Айтида предусматривает механизмы резервирования товаров на складах для различных целей. Зарезервированный товар может находиться на складе, а может и не находиться, тогда резерв распространится на ближайшую поставку. Резервирование товара может производиться для контрагента или для склада получателя. В стандартной конфигурации резерв для контрагента создается документами Заказ от покупателя и Счет исходящий. Для другого склада – документом Резервирование ТМЦ. Резерв на складе имеет срок окончания резерва, по истечении которого резерв считается аннулированным. Резерв может быть отменен другим документом Заказ от покупателя/Резервирование ТМЦ с указанием операции отмена резерва, либо в момент отгрузки расходной накладной контрагенту или внутренним перемещением на склад получатель.

Таблица reg002.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). '+' увеличивает резерв, '-' уменьшает
account_	char(3)	Код модели учета.
nn	char(10)	Внутренний код ТМЦ
sklad	char(3)	Код склада, на котором резервируется ТМЦ.
dat_end	datetime	Срок окончания резерва.
kolp	float	Количество зарезервированных ТМЦ
realkolp	float	Количество, реально списанное с резервов.
rtype	char(1)	не используется.
code_s	char(3)	Код документа, с которого необходимо списать резерв.
ic_s	int	Идентификатор документа, с которого необходимо списать резерв.
kolp_	float	Остаток несписанного резерва.
sklad_d	char(3)	Код склада, для которого зарезервированы ТМЦ.
client	char(7)	Код контрагента, для которого зарезервированы ТМЦ.
transtime	datetime	Дата и время создания записи

### Примечание.

1. Для каждой записи добавившей резерв, система отслеживает количество списанного с нее резерва. Это делается для обеспечения возможности оперативной оценки списания резервов по тому или иному документу. Для документов, которые списывают ТМЦ с резервов, например расходная накладная или внутреннее перемещение, система рассчитывает сколько на самом деле было списано резервов с того или иного документа резервирования. Это делается для возможности оперативной оценки того, сколько ТМЦ было списано из резерва, а сколько из свободного количества.
2. В стандартной конфигурации документы влияют на регистр учета резервов ТМЦ, если
  - a. в параметрах системы указаны вести учет резервов ТМЦ,
  - b. склад, на котором производится резервирование, находится в текущем филиале,
  - c. при списании резерва документами расходная накладная и внутреннее перемещение, необходимо указание документа, с которого списывается резерв.

Для расчета количества зарезервированных ТМЦ на складе необходимо использовать хранимую функцию:

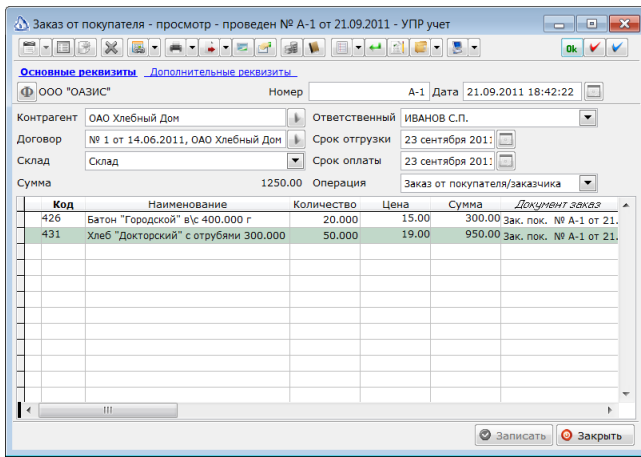
*fn\_calcreserv (@nn char(10), @date datetime, @sklad char(3), @account\_ char(3), @code\_s char(3), @ic\_s int)*

Для расчета количества зарезервированных ТМЦ с учетом всех подчиненных ТМЦ необходимо использовать функцию:

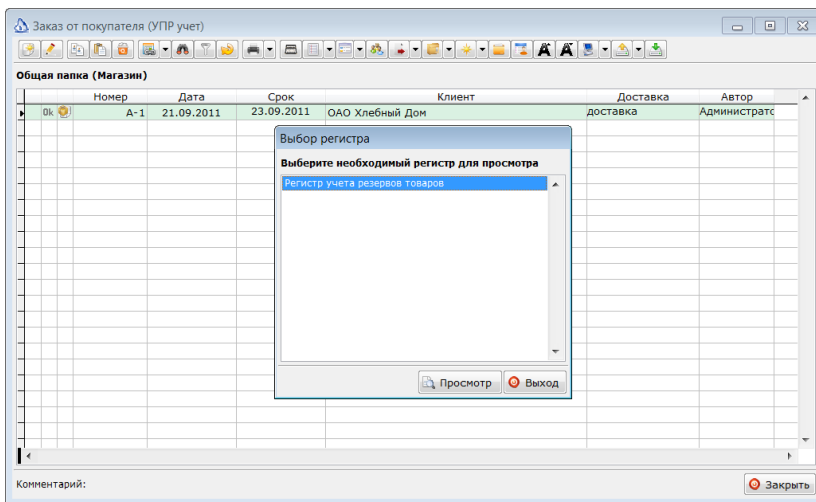
*fn\_calcreservsubres (@nn char(10), @date datetime, @sklad char(3), @account\_ char(3))*

## Примеры заполнения регистра учета резервов ТМЦ и использования функций.

В качестве примера заведем в систему документ Заказ от покупателя, как указано на рисунке.

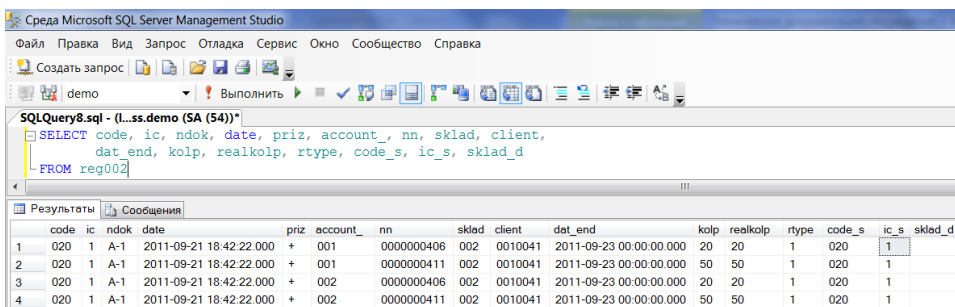


Данный документ влияет только на один регистр – регистр учета резервов ТМЦ:



Выполним запрос для просмотра содержимого регистра *reg002*. Т.к. в системе только один влияющий на этот регистр, то никаких дополнительных условий не указываем.

```
SELECT code, ic, ndok, date, priz, account_, nn, sklad, client,
       dat_end, kolp, realkolp, rtype, code_s, ic_s, sklad_d
FROM reg002
```



Результат содержит строки для двух моделей учета – управленческого и бухгалтерского. Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	020	Код типа документа Заказ от покупателя
ic	1	Идентификатор документа. Равно полю identity_column в таблице spr020
ndok	A-1	Номер документа

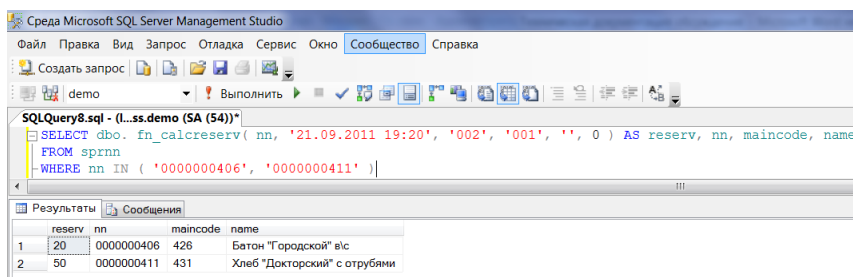
date	2011-09-21 18:42:22.000	Дата документа. В стандартной конфигурации, все документы создают записи в этом регистре с датой равной дате документа.
priz	+	Текущая запись добавляет ТМЦ к резерву.
account_	001	Код модели управленческого учета.
nn	0000000406	Код резервируемого ТМЦ.
sklad	002	Код склада, на котором резервируется ТМЦ.
client	0010041	Код контрагента, для которого резервируется ТМЦ. Для документов, резервирующих ТМЦ для передачи на другой склад, это поле будет пустое.
dat_end	2011-09-23 00:00:00.000	Срок окончания резерва.
kolp	20	Зарезервированное количество.
realkolp	20	Для записей, добавляющих резерв, это количество равно полю kolp.
rtype	1	Тип резерва. В стандартной конфигурации всегда '1'.
code_s	020	Для документов, добавляющих резерв, это тип самого документа. Для документов снимающих резерв, это поле содержит тип документа, добавившего резерв.
ic_s	1	Для документов, добавляющих резерв, это идентификатор самого документа. Для документов снимающих резерв, это поле содержит идентификатор документа, добавившего резерв.
sklad_d		Для документов, резервирующих ТМЦ для передачи на другой склад, это поле содержит код склада получателя.

*Функция fn\_calcreserv – расчет количества ТМЦ зарезервированных на складе.*

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток.
@date	datetime	Дата и время, на которые необходимо произвести расчет.
@sklad	char(3)	Код склада, на котором необходимо определить остаток.
@account_	char(3)	Код модели учета для которой производится расчет.
@code_s	char(3)	Код документа резерва
@ic_s	int	Идентификатор документа резерва.

Выполним следующий запрос, в качестве условия укажем список ТМЦ, который был зарезервирован в примере:

```
SELECT dbo. fn_calcreserv( nn, '21.09.2011 19:20', '002', '001', '', 0 ) AS reserv,
      nn, maincode, name
FROM sprnn
WHERE nn IN ( '0000000406', '0000000411' )
```



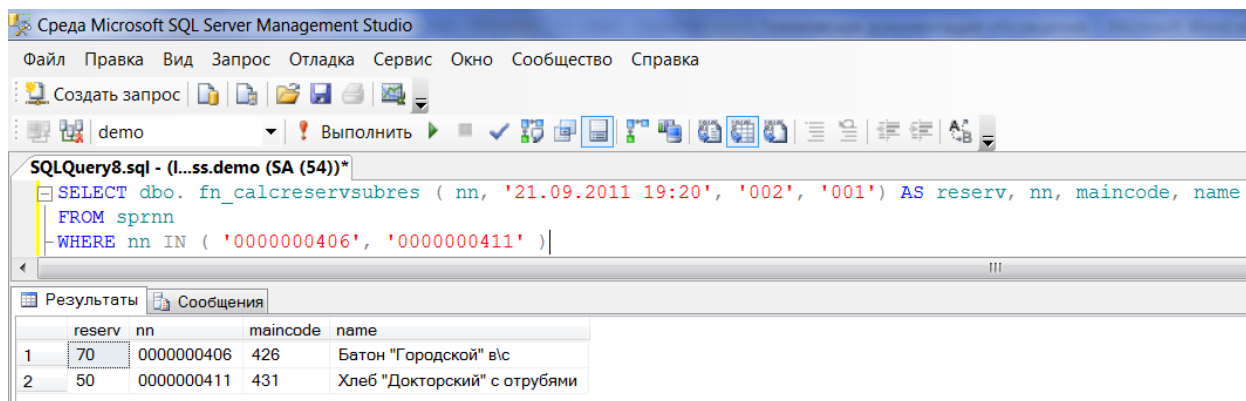
В первом столбце результата отображен зарезервированное количество ТМЦ. Следует отметить, что если в качестве параметра @date передать дату и время позже 23.09.2011 00:00, то результат будет 0, т.к. функция проверяет срок резерва на складе.

*Функция fn\_calcreservsubres – расчет количества зарезервированных ТМЦ с учетом всех подчиненных ТМЦ на складе.*

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которого необходимо рассчитать остаток.
@date	datetime	Дата и время, на которые необходимо произвести расчет.
@sklad	char(3)	Код склада, на котором необходимо определить остаток.
@account_	char(3)	Код модели учета для которой производится расчет.

Выполним следующий запрос, в качестве условия укажем список ТМЦ, который был резервирован в примере:

```
SELECT dbo. fn_calcresevsres ( nn, '21.09.2011 19:20', '002', '001') AS reserv,  
       nn, maincode, name  
FROM sprnn  
WHERE nn IN ( '0000000406', '0000000411' )
```



Обратите внимание, для ТМЦ с кодом 0000000406 рассчитано количество 70, т.к. у нее в подчинении находится ТМЦ с кодом 0000000411.

## Регистр учета денежных средств.

Регистр учета денежных средств предназначен для сбора информации о движении наличных и безналичных денежных средств в различных валютах. В стандартной конфигурации движение денежных средств осуществляется следующими документами:

1. Банковская выписка (код 012).
2. Приходный кассовый ордер (код 023).
3. Расходный кассовый ордер (код 016).
4. Приход наличных денег (код 013).
5. Расход наличных денег (код 017).
6. Товарный чек (код 004).
7. Авансовый отчет (код 044).

Основным разрезом ведения учета, является денежный карман – аналог склада для хранения денежных средств. Денежные карманы могут быть наличные и безналичные. Денежные карманы привязаны к филиалу, в котором они находятся.

Таблица reg003.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). '+' увеличивает сумму в денежном кармане, '-' уменьшает ее.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
kassakod	char(3)	Код денежного кармана, для осуществления движения.
cur	char(3)	Код валюты операции.
summa	float	Сумма операции

### Примечание.

1. В стандартной конфигурации документы влияют на регистр учета денежных средств, если
  - a. денежный карман, для которого производится движение, находится в текущем филиале;
  - b. если расходный кассовый ордер или расход наличных денег осуществляют перемещение денежных средств между денежными карманами (в качестве получателя указан денежный карман), то денежные средства приходятся в денежный карман получатель, только если установлен признак Приход и указана дата прихода. При этом операция проводится датой прихода денежных средств;
  - c. если приходный кассовый ордер, приход наличных денег или товарный чек осуществляют перемещение денежных средств между денежными карманами (в качестве отправителя указан денежный карман), то операции прихода и расхода денежных средств осуществляются датой документа;
  - d. банковская выписка осуществляет движение безналичных денежных средств, если денежный карман, указанный в расчетном счете банковской выписки находится в текущем филиале;
  - e. авансовый отчет влияет на регистр денежных средств, если сотрудник сдает деньги в кассу, при этом в документе должен быть указан тип получателя – денежный карман, а в поле получатель должен быть выбран наличный денежный карман, находящийся в текущем филиале.
2. Банковская выписка осуществляет движение денежных средств одной операцией, а не по каждой строке выписки.
3. Для получения информации из регистра учета денежных средств предусмотрены следующие функции:
  - a. Для расчета суммы остатка денежных средств в денежном кармане, необходимо использовать хранимую функцию:  
*fn\_calcmoney (@kassacode char(3), @cur char(3), @date datetime, @account\_ char(3))*

- b. Для расчета остатков денежных средств по всем карманам и всем валютам, необходимо использовать функцию:  
`fn_calcmoney_all (@date datetime, @account_char(3))`
- c. Для расчета оборотов денежных средств за период, в разрезе денежных карманов и валют, необходимо использовать функцию:  
`fn_calcmoney_turnover (@kassacode, @cur char(3), @date_min datetime, @date_max datetime, @account_char(3))`

### Примеры заполнения регистра учета денежных средств и использования функций.

В качестве примера заведем в систему документ Приходный кассовый ордер, как указано на рисунке.

Проведем данный документ и посмотрим, на какие регистру влияет данный документ.

Выполним запрос для просмотра содержимого регистра `reg003`. Т.к. в системе только один влияющий на этот регистр, то никаких дополнительных условий не указываем.

```
SELECT code, ic, ndok, date, priz, cur, summa, kassakod, account_
FROM reg003
```

code	ic	ndok	date	priz	cur	summa	kassakod	account_
1	023	29	A-1	2011-09-22 00:00:00.000	+	USD 1000	004	001
2	023	29	A-1	2011-09-22 00:00:00.000	+	USD 1000	004	002

Результат содержит строки для двух моделей учета – управленческого и бухгалтерского. Рассмотрим первую строку результата.

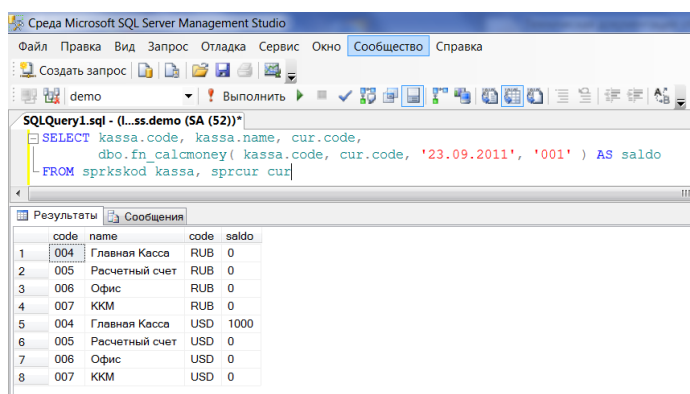
Имя поля	Значение	Комментарий
code	023	Код типа документа Приходный кассовый ордер
ic	1	Идентификатор документа. Равно полю identity_column в таблице spr023
ndok	A-1	Номер документа
date	2011-09-22 00:00:00.000	Дата документа. В стандартной конфигурации, все документы создают записи в этом регистре с датой равной дате документа.
priz	+	Добавление суммы в денежный карман.
cur	USD	Валюта денежных средств. В стандартной конфигурации все документы устанавливают это поле равным валюте документа.
summa	1000	Сумма оприходованных денежных средств в указанной валюте.
kassakod	004	Код денежного кармана, в который оприходованы денежные средства.
account_	001	Код модели управленческого учета.

Функция *fn\_calcmoney* расчет суммы остатка денежных средств в денежном кармане.

Имя параметра	Тип параметра	Комментарий
@kassacode	char(3)	Код денежного кармана, для которого необходимо определить остаток.
@cur	char(3)	Код валюты, для которой необходимо определить остаток.
@date	datetime	Дата и время для расчета остатка
@account_	char(3)	Код модели учета для расчета.

Выполним запрос для формирования списка остатков по денежным карманам и валютам.

```
SELECT kassa.code, kassa.name, cur.code,
       dbo.fn_calcmoney( kassa.code, cur.code, '23.09.2011', '001' ) AS saldo
FROM sprkskod kassa, sprcur cur
```

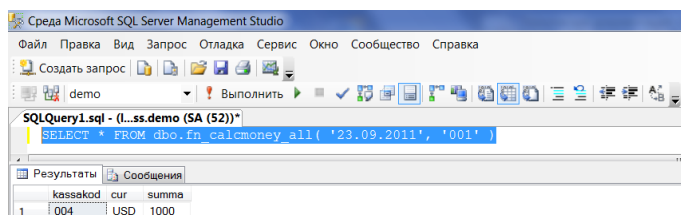


Функция *fn\_calcmoney\_all* расчет остатков денежных средств по всем карманам и валютам.

Имя параметра	Тип параметра	Комментарий
@date	datetime	Дата, на которую необходимо произвести расчет
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. В отличии от предыдущего примера, эта функция возвращает список только тех денежных карманов и валют, по которым было движение до указанной даты.

```
SELECT * FROM dbo.fn_calcmoney_all( '23.09.2011', '001' )
```



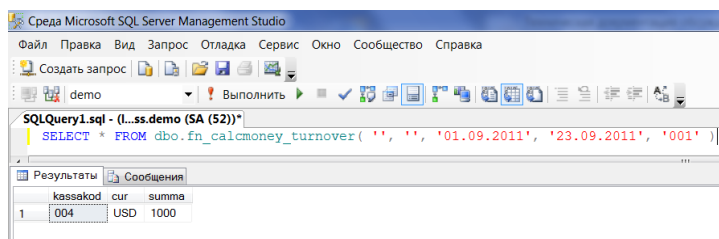


Функция *fn\_calcmoney\_turnover* расчет оборотов денежных средств за период, в разрезе денежных карманов и валют.

Имя параметра	Тип параметра	Комментарий
@kassacode	char(3)	Код денежного кармана, для которого необходимо определить остаток.
@cur	char(3)	Код валюты, для которой необходимо определить остаток.
@date_min	datetime	Дата начала расчетного периода.
@date_max	datetime	Дата окончания расчетного периода.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Функция возвращает оборот по тем денежным карманам и валютам, по которым был оборот в указанном периоде.

```
SELECT * FROM dbo.fn_calcmoney_turnover( '', '', '01.09.2011', '23.09.2011', '001' )
```



## Регистр учета взаиморасчетов с контрагентами.

Регистр учета взаиморасчетов с контрагентами предназначен для сбора следующей информации.

1. Информации о влиянии документов на взаиморасчеты в указанной валюте. В стандартной конфигурации – в валюте документа.
2. Закрытие взаиморасчетов при сопоставлении разнонаправленных документов.
3. Информации суммовых разниц, возникающих при сопоставлении документов в разных валютах. Суммовые разницы рассчитываются в валюте учета.
4. Информации о курсовых разниц, возникающих при переоценке дебиторской и кредиторской задолженностей. Курсовые разницы рассчитываются в валюте учета.

В стандартной конфигурации движения в данном регистре могут создавать все документы, в которых присутствует контрагент. Если используемая конфигурация позволяет вести учет взаиморасчетов в разрезе договоров, то помимо контрагента, взаиморасчеты ведется в разрезе договоров.

Таблица reg004.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). Записи с признаком '+' увеличивает соответствующую задолженность. Записи с признаком '-' уменьшает ее.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
client	char(7)	Код с контрагентов, с которым ведутся взаиморасчеты.
contract	char(10)	Код договора, в разрезе которого должны вестись взаиморасчеты.
firm	char(7)	Код организации, которая ведет взаиморасчеты с контрагентом.
cur	char(3)	Код валюты, в которой фиксируется взаиморасчета.
cur_rate	float	Курс валюты взаиморасчета к валюте учета.
proc_	float	Дополнительный процент к курсу валюты.
summa_po	float	Сумма выплаченных контрагенту денежных средств.
summa_pi	float	Сумма полученных от контрагента денежных средств.
summa_o	float	Сумма отгруженных контрагенту товаров и услуг.
summa_i	float	Сумма полученных от контрагента товаров и услуг.
adjustment	float	Сумма переоценки задолженности контрагента/организации.
adjustment_koef	float	Коэффициент переоцениваемой задолженности.
ctype	char(1)	Тип операции. Пустая – основной тип начисления задолженности. 'D' – сопоставление - оплата, закрытие основной задолженности. 'M' – сопоставление – отгрузка/оприходование. 'G' – Суммовая разница, возникшая при сопоставлении. 'A' – Курсовая разница, возникшая при переоценке.
cross_ic	int	Идентификатор связи, которая создала запись.

### Примечание.

1. В стандартной конфигурации системы Айтида применяются следующие правила заполнения полей с суммами:
  - a. денежные исходящие документы (когда организация платит контрагенту) заполняют поле `summa_po`;
  - b. денежные входящие документы (когда контрагента платит организации) заполняют поле `summa_pi`;
  - c. документы поставки ТМЦ и акты услуг входящие заполняют поле `summa_i`;
  - d. документы отгрузки ТМЦ и акты услуг исходящие заполняют поле `summa_o`;

Все указанные суммы заносятся со знаком плюс и признаком `priz` равным '+'. Таким образом, чтобы узнать задолженность контрагента необходимо использовать следующую формулу:

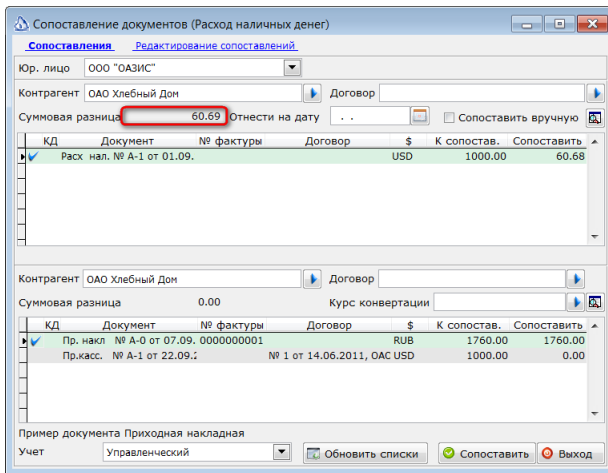
$summa_o + summa_po - summa_i - summa_pi$ . Если результат получился положительный, то это значит, что контрагент должен организации.

2. В стандартной конфигурации документы влияют на регистр взаиморасчетов с контрагентами если
  - a. поле контрагента в документе не пустое;
  - b. код филиала в карточке контрагента соответствует коду текущего филиала;
  - c. установлен признак «вести взаиморасчеты» в карточке контрагента;
  - d. это *приходная накладная* и тип документа не равен БОНУС;
  - e. это *расходная накладная* и тип расхода не равен КОМИССИЯ;
  - f. это *платежное поручение* или *аванс полученный* и не стоит признак «не влиять на взаиморасчеты»;
  - g. это банковская выписка и в строке с контрагентом установлен флаг «влиять на взаиморасчеты»
3. В момент создания связи *сопоставление-оплата* или *сопоставление-отгрузка/оприходование* обрабатывает триггер на добавление записи в таблице **crosslink**. Этот триггер создает в регистре учета взаиморасчетов с контрагентами записи с типом (поле **ctype**) 'D' на сумму сопоставления. Эти записи создаются в той же валюте и с тем же курсом, что и исходные записи документа, но с признаком (поле **priz**) равным '-'. Дата устанавливается равной дате сопоставления (поле **date\_link** в таблице **crosslink**). Таким образом, закрываются взаиморасчеты по сопоставляемым документам. В случае, если происходило сопоставление документов в валюте отличной от валюты учета, то одновременно рассчитывается суммовая разница для каждого из таких сопоставляемых документов. Эта суммовая разница добавляется в регистр учета взаиморасчетов в виде записи, у которой валюта устанавливается равной валюте учета, дата – дате сопоставления, признак равным '-', а тип равным 'G'. **Важно.** Записи, созданные *сопоставлениями*, не удаляются из регистра при отмене проведения документа. Эти записи удаляются только при удалении связи *сопоставление*, которой они были созданы.
4. Система Айтида предусматривает проведение периодической процедуры переоценки дебиторской и кредиторской задолженностей. Переоценке подлежат задолженности в валюте отличной от валюты учета и не закрытые полностью на момент проведения переоценки. Процедура переоценки рассчитывает курсовую разницу на остаток задолженности и добавляет по каждому не закрытому документу запись в регистр взаиморасчетов на сумму переоценки (сумма записывается в поле **adjustment**) валюте учета. Тип записи устанавливается равным 'A'. Данные переоценки будут сторнированы при сопоставлении переоцененных документов.

### Примеры заполнения регистра учета взаиморасчетов с контрагентами.

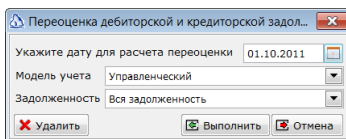
В качестве примера заведем в систему документ Расход наличных денег, как указано на рисунке.

В справочнике валют укажем курс валюты для доллара США на 01.09.2011 равный 28 рублям и на 05.09.2011 равный 29 рублям. Таким образом, указанный документ соответствует 28 000 рублям. Теперь выполним сопоставление этого документа с приходной накладной из предыдущих примеров следующим образом:

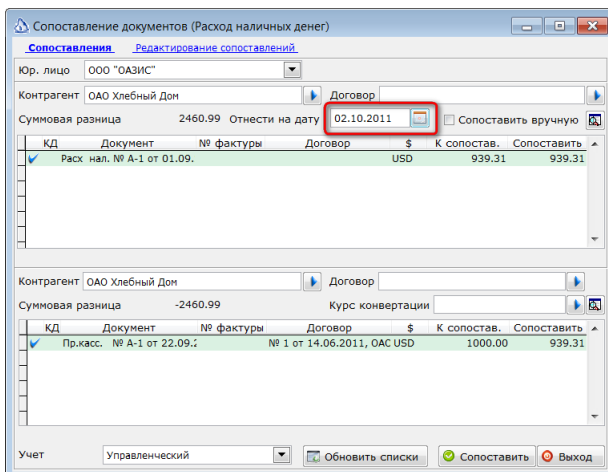


Система произвела расчет следующим образом. Приходная накладная сопоставлена полностью, т.к. ее сумма значительно меньше суммы расхода наличных денег. Расход наличных денег сопоставился на сумму равную сумме приходной накладной умноженной на курс доллара США на дату приходной накладной:  $1760 / 29 = 60,689655$  (в таблице отображаются только два знака после запятой, но в базу данных будет записано число со всеми знаками). Кроме того, была рассчитана суммовая разница на сумму 60 рублей 69 копеек:  $(1760 / 29) * (29 - 28) = 60,69$ .

Кроме этого, произведем переоценку задолженности на 01.10.2011. Перед этим установим курс валюты доллар США на 01.10.2011 равным 32 рублям.



И в заключение, сопоставим расход наличных денег с приходным кассовым ордером из предыдущего примера, установив дату отнесения на 02.10.2011.



Рассмотрим, как все эти данные повлияли на регистр взаиморасчетов с контрагентами:

```
SELECT code, ic, ndok, priz, account_, client, contract,
       firm, cur, cur_rate, summa_po, summa_pi, summa_o,
       summa_i, adjustment, adjustment_koef, ctype, cross_ic
FROM reg004 WHERE code='017' and ic= 2
```

Следует отметить, что переоценка проводилась по бухгалтерскому учету, поэтому ее нет в управленческом учете.

Среда Microsoft SQL Server Management Studio

SQLQuery1.sql - (L...ss.demo (SA (53))\*

```
SELECT code, ic, ndok, date, priz, account, client, contract,
       firm, cur, cur_rate, summa_po, summa_pi, summa_o,
       summa_i, adjustment, adjustment_koef, ctype, cross_ic
FROM reg004 where code='017' and ic= 2 ORDER BY account_ DESC, cross_ic
```

code	ic	ndok	date	priz	account	client	contract	firm	cur	cur_rate	summa_po	summa_pi	summa_o	summa_i	adjustment	adjustment_koef	ctype	cross_ic
017	2	A-1	2011-09-01 14:43:51.000	+	002	0010041		0010001	USD	0,03571428571	1000	0	0	0	0	0	A	0
017	2	A-1	2011-10-01 00:00:00.000	+	002	0010041		0010001	RUB	1	0	0	0	0	3757,23999684392	0,93931	A	0
017	2	A-1	2011-09-01 14:43:51.000	-	002	0010041		0010001	USD	0,03571428571	60,69	0	0	0	0	0	D	1209
017	2	A-1	2011-09-01 14:43:51.000	-	002	0010041		0010001	RUB	1	60,6896550024831	0	0	0	0	0	G	1209
017	2	A-1	2011-10-02 00:00:00.000	-	002	0010041		0010001	USD	0,03571428571	939,31	0	0	0	0	0	D	1213
017	2	A-1	2011-10-02 00:00:00.000	-	002	0010041		0010001	RUB	1	2460,99222947691	0	0	0	0	0	G	1213
017	2	A-1	2011-10-02 00:00:00.000	-	002	0010041		0010001	RUB	1	0	0	0	0	3757,23999684392	0,93931	A	1213
017	2	A-1	2011-09-01 14:43:51.000	+	001	0010041		0010001	USD	0,03571428571	1000	0	0	0	0	0		0
017	2	A-1	2011-09-01 14:43:51.000	-	001	0010041		0010001	USD	0,03571428571	60,69	0	0	0	0	0	D	1207
017	2	A-1	2011-09-01 14:43:51.000	-	001	0010041		0010001	RUB	1	60,6896550024831	0	0	0	0	0	G	1207
017	2	A-1	2011-10-02 00:00:00.000	-	001	0010041		0010001	USD	0,03571428571	939,31	0	0	0	0	0	D	1215
017	2	A-1	2011-10-02 00:00:00.000	-	001	0010041		0010001	RUB	1	2460,99222947691	0	0	0	0	0	G	1215

**Примечание.** Рассмотрим более подробно полученный результат по бухгалтерскому учету. По управленческому учету все аналогично, за исключением переоценок.

1. При первичном проведении документа была создана основная запись на сумму 1000 USD. Выделена зеленым цветом.
2. Было выполнено сопоставление с приходной накладной на сумму 1760 рублей или 60,69 USD. Выделено желтым цветом.
3. Была произведена переоценка остатка не сопоставленной суммы 939,21 USD по формуле  $(32 - 28) * 939,31 = 3757,24$  рублей. Выделено синим цветом.
4. Было выполнено сопоставление с приходным кассовым ордером на остток суммы 939,31. При этом была сторнирована (записана с признаком priz = '-') ранее выполненная переоценка на сумму 3757,24 рублей и начислена суммовая разница, рассчитанная по формуле  $(30,62 - 28) * 939,31 = 2460,99$  рублей, где 30,62 – курс доллара США на дату приходного кассового ордера, 28 – курс доллара США на дату расхода наличных денег. Стоит отметить, что, так как оба документа выписаны в одной валюте (долларах США), то аналогичная суммовая разница, только с обратным знаком, была начислена и для документа Приходный кассовый ордер.

Рассмотрим более подробно первую строку результата выборки

Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	017	Код типа документа Расход наличных денег.
ic	2	Идентификатор документа. Равно полю identity_column в таблице spr017
ndok	A-1	Номер документа
date	2011-09-01 14:43:51.000	Дата документа. Для остальных записей это дата сопоставления, возможно указанная вручную или дата переоценки, указанная в форме переоценки.
priz	+	Добавление соответствующей суммы во взаиморасчеты. При сопоставлении, записи добавляются с признаком '+-'
account_	002	Код модели учета, по которой произведен взаиморасчет.
client		Код контрагента, с которым произведен взаиморасчет.
contract		Код договора, по которому произведен взаиморасчет.
firm		Код организации из документа.
cur	USD	Валюта взаиморасчета. Для основной записи и для закрывающих записей с типом 'D' значение поля равно валюте документа. Для суммовой разницы и переоценки равно валюте учета.
cur_rate	0,03571428571	Курс валюты документа к основной валюте учета. В данном случае означает, что один рубль равен указанному количеству долларов США, т.е. для получения суммы документа в валюте учета необходимо 1000

		USD / 0,03571428571 = 28 000 рублей.
summa_po	1000	Сумма взаиморасчета – «выдано денежных средств»
summa_pi	0	Сумма взаиморасчета – «получено денежных средств»
summa_o	0	Сумма взаиморасчета – «выдано ТМЦ»
summa_i	0	Сумма взаиморасчета – «получено ТМЦ»
adjustment	0	Сумма переоценки.
adjustment_koef	0	Коэффициент переоцененной задолженности от общей суммы взаиморасчета. В данном случае рассчитан по формуле $(1000 - 60,69)/1000 = 0,93931$
ctype		Тип записи. Пусто – основная запись, D – сопоставление, G – суммовая разница, A – переоценка задолженности.
cross_ic	0	Идентификатор записи из таблицы сопоставлений (crosslink), для которой создана данная запись.

## Регистр учета взаиморасчетов между филиалами.

Регистр учета взаиморасчетов между филиалами предназначен для сбора информации о задолженностях, возникающих между филиалами при перемещении между ними ТМЦ и денежных средств.

В стандартной конфигурации, движения в данном регистре могут создавать следующие документы:

1. **Внутреннее перемещение**, если филиал отправитель или филиал получатель не являются текущим филиалом.
2. **Банковская выписка**, если в строке многострочной части документа указан код филиала отличный от текущего филиала.
3. **Денежные документы (приходный кассовый ордер, расходный кассовый ордер, приход наличных денег, расход наличных денег)**, если филиал отправитель или филиал получатель не являются текущим филиалом.
4. **Документы, перемещающие ценные бумаги**, если филиал отправитель или филиал получатель не являются текущим филиалом.

Таблица reg005.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). Записи с признаком '+' увеличивает соответствующую задолженность. Записи с признаком '-' уменьшает ее.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
branch	char(3)	Код филиала, с которым произведен расчет.
summa	float	Сумма расчета.

**Примечание.** Все суммы расчетов между филиалами хранятся в основной валюте учета. Для получения сальдо по задолженности указанного филиала, необходимо воспользоваться хранимой функцией:

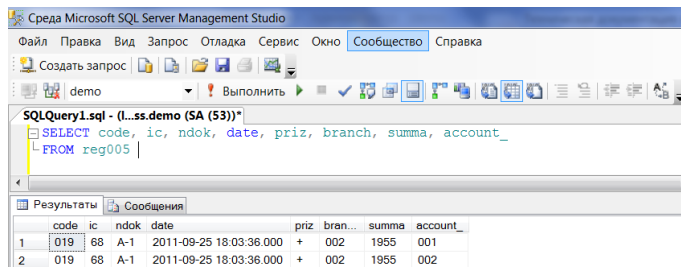
1. Получение остатка по филиалу:  
*fn\_calcbbranch\_saldo(brach char(3), @date datetime, @account char(3))*

## Примеры заполнения регистра учета расчетов между филиалами и использования функции.

Для примера создадим в справочнике складов новый склад. Назовем его Склад филиала и укажем, что он находится в Филиале:

Создадим новый документ внутреннего перемещения и переместим ранее оприходованный товар на Склад филиала. Проведем этот документ и посмотрим, как она отразилась на регистре учета расчетов между филиалами. Выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, branch, summa, account_
FROM reg005
```



Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	019	Код типа документа Внутреннее перемещение.
ic	2	Идентификатор документа. Равно полю identity_column в таблице spr019
ndok	A-1	Номер документа
date	2011-09-01 14:43:51.000	Дата документа. Все документы, кроме внутреннего перемещения создают записи датой документа. Документ внутреннее перемещение создает записи списания задолженности датой документа, а начисления задолженности датой прихода.
priz	+	Добавление соответствующей суммы в расчеты.
branch	002	Код филиал, у которого образовалась задолженность.
summa	1955	Сумма задолженности. Суммы рассчитываются в учетных ценах склада получателя.
account_	002	Код модели учета, по которой произведен расчет.

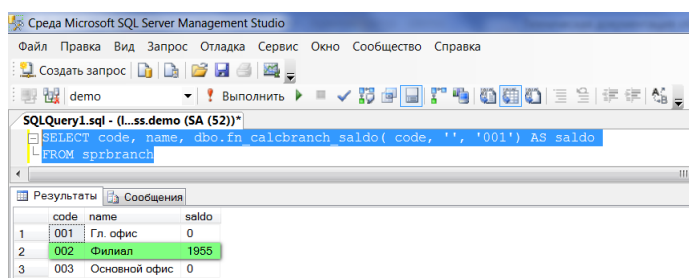
**Примечание.** При перемещении денежных средств задолженность возникает на сумму документа перемещения. Если перемещается сумма в валюте отличной от валюты учета, то задолженность возникает по курсу указанному в документе.

*Функция fn\_calcbranch\_saldo расчет задолженности филиала на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@branch	char(3)	Код филиала, для которого рассчитывается задолженность.
@date	datetime	Дата расчета задолженности.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Функция возвращает оборот по тем денежным карманам и валютам, по которым был оборот в указанном периоде.

```
SELECT code, name, dbo.fn_calcbranch_saldo( code, '', '001') AS saldo
FROM sprbranch
```





## Регистр учета номеров партий ТМЦ.

Регистр учета номеров партий ТМЦ предназначен для сбора информации о движении (поступлении и списании) ТМЦ в разрезе номеров партий. Номера партий появляются в системе в момент поступления ТМЦ, оформляемого документами прихода. В системе предусмотрено как автоматическое формирование номеров партий, так и “ручное” указание номеров партий при вводе документов. При списании, если пользователь не указал требуемые номера партий, система производит автоматическое распределение номеров среди партий с не нулевым остатком в порядке FIFO. Если пользователь указал конкретные номера партий, то система производит списание с указанных номеров и в случае нехватки требуемого количества, отказывает в этом.

Для активации работы с номерами партий необходимо в параметрах системы установить признак “Вести учет ТМЦ в разрезе партий”. Так же необходимо хотя бы в одной карточке справочника **окружений** установить признак “Вести учет партий ТМЦ”. Рекомендуется указывать необходимость ведения учета в разрезе номеров партий только для одной модели учета. В противном случае вероятно возникновение ситуации, при которой система не сможет провести какой-либо документ списания, т.к. в одной модели учета будут в наличии номера партий для списания, а в другой – нет. В системе не предусмотрена возможность появления отрицательного остатка ТМЦ в разрезе партий.

Система Айтида предусматривает вариант учета номеров партий в разрезе Организаций, что позволяет разделить учет номеров партий между собственными юридическими лицами.

Учет комиссионной торговли возможен только при использовании учета номеров партий ТМЦ. Система Айтида предусматривает возможность указания приоритета списания собственного или комиссионного товара.

В стандартной конфигурации следующие документы приходят номера партий.

1. Приходная накладная.
2. Оприходование излишков ТМЦ.
3. Возврат от покупателей.
4. Инвентаризация ТМЦ.
5. Выпуск и комплектация.
6. Внутреннее перемещение.
7. Пересортица ТМЦ.
8. Демонтаж ОС.

Списание производится следующими документами.

1. Расходная накладная.
2. Списание ТМЦ.
3. Инвентаризация ТМЦ.
4. Документ кассовой смены.
5. Выпуск и комплектация.
6. Внутреннее перемещение.
7. Пересортица ТМЦ.
8. Формирование стоимости объекта ОС.
9. Возврат поставщику.

Таблица reg006.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). Записи с признаком '+' приходят новый номер партии ТМЦ. Записи с признаком '-' списывают номер партии ТМЦ.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
spec_ic	int	Идентификатор строки многострочной части документа, к которой относится строка в регистре.
nn	char(10)	Код ТМЦ
bc	varchar(50)	Номер учитываемой партии.
bc_date	datetime	Дата, связанная с номером партии. Например, срок годности или дата производства алкоголя и т.п.
firm	char(7)	Код организации, если учет ведется в разрезе организаций.
kolp	float	Приходящее или списываемое количество ТМЦ.
kolp_in	float	Для записей приходящих номера партий содержит не распределенное количество. Поле используется для организации схемы FIFO.
fauto	bit	Признак того, что необходимо автоматическое распределение номеров списываемых партий (пользователь не указывал номера партий в документах списания).
fsimple	bit	Признак, того ведется учет номеров партий по упрощенной схеме FIFO.
fbyfirm	bit	Признак того, что необходимо вести учет в разрезе организаций.
type	int	Тип прихода / списания (1 - Закуп/Продажа, 2 - Комиссия, 3 - Бонус, 4 - Прочее).
ordertype	int	Необходимый порядок сортировки приходов при списании (0 - без учета типа прихода, 1 - Закуп-Комиссия-Бонус-Прочее, 2-...).
_ic	int	Ссылка на запись в регистре reg006, по которой производилось списание (используется в схемах FIFO).
_date	datetime	Для списываемого прихода номера партии. Поле используется для организации схемы FIFO.
blockfifo	bit	Блокировка триггеров при выполнении внутренних процедур перераспределения номеров партий.

**Примечание.**

- В системе Айтида реализованы механизмы автоматического перераспределения номеров партий ТМЦ при оформлении операций "задним числом". При оформлении прихода задним числом все последующие расходы подвергаются перераспределению с целью соблюдения принципа FIFO. При добавлении расхода задним числом, так же производится перераспределение всех последующих расходов. При этом может возникнуть ситуация, при которой ранее списанные серийные номера не смогут быть списаны повторно. Это приведет к ошибке и к отказу в проведении документа. Пример.
  - Оформим приходную накладную на ТМЦ А с номером партии НомерА, количеством 1 и датой 01.01.2001.
  - Затем, оформим приходную накладную на ТМЦ А с номером партии НомерВ, количеством 1 и датой 02.01.2001.
  - Оформим расходную накладную № 2 на ТМЦ А, с количеством 1 и датой 04.01.2001. Укажем в документе, что хотим списать номер партии НомерА
  - Оформим расходную накладную № 1 на ТМЦ А, с количеством 1 и датой 03.01.2001. Номер партии указывать не будем, для того, чтобы система автоматически распределила номера.

Система не сможет провести этот документ, хотя, казалось бы, количества товара на складе хватает для выполнения операции. Но, система будет пытаться перераспределить номера партий между новой последовательностью документов списания. При этом партия с номером НомерА будет списана документом № 1 от 03.01.2001. В документе № 2 указан конкретный номер партии НомерА, но этот номер уже был списан документом № 1. Система выдаст сообщение о нехватке номеров партий для распределения и отменит операцию.
- Полное и точное соблюдение принципа FIFO достаточно ресурсоемкая задача, особенно при проведении документов задним числом, когда требуется перераспределение более поздних документов. Поэтому, в

системе Айтида предусмотрен вариант организации списания номеров партий по “упрощенной” схеме. При этой схеме при наличии возможности система не перераспределяет все последующие расходы. Так в предыдущем примере, при проведении накладной №1 задним числом, система бы списала его с номером партии НомерВ. При этом была бы нарушена строгость списания по схеме FIFO, но оба расходных документа удалось бы провести, при этом количество затраченных ресурсов на операцию было бы существенно меньше. Поэтому, если условия позволяют, рекомендуется использовать именно “упрощенную” схему списания.

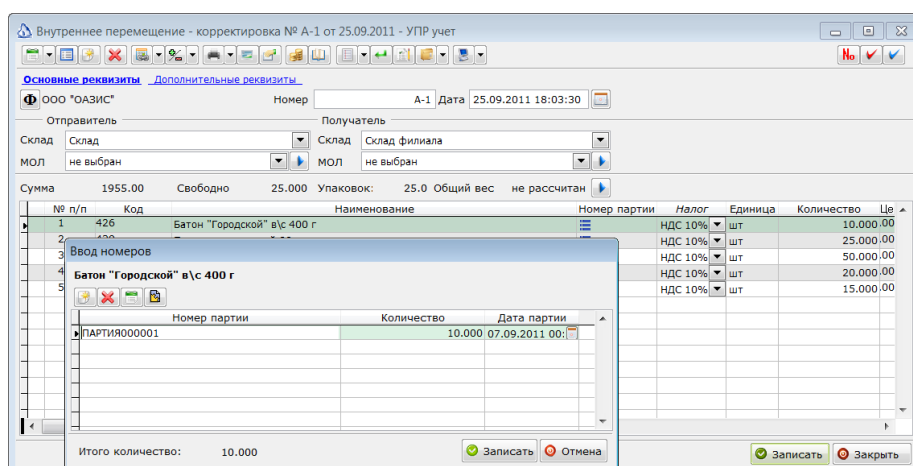
3. При установленном признаке “Сохранять номера партий в документах” в карточке окружения, в момент проведения документа списания, автоматически распределённые номера партий будут записаны в таблицу `barcodes2`. При этом будет сохранен идентификатор записи регистра (поле `reg006_ic` в таблице `barcodes2`), который создал эту запись. При удалении записи из регистра `reg006`, автоматически добавленные записи будут удалены.

### Примеры заполнения регистра учета номеров партий.

Рассмотрим ранее созданные документы:

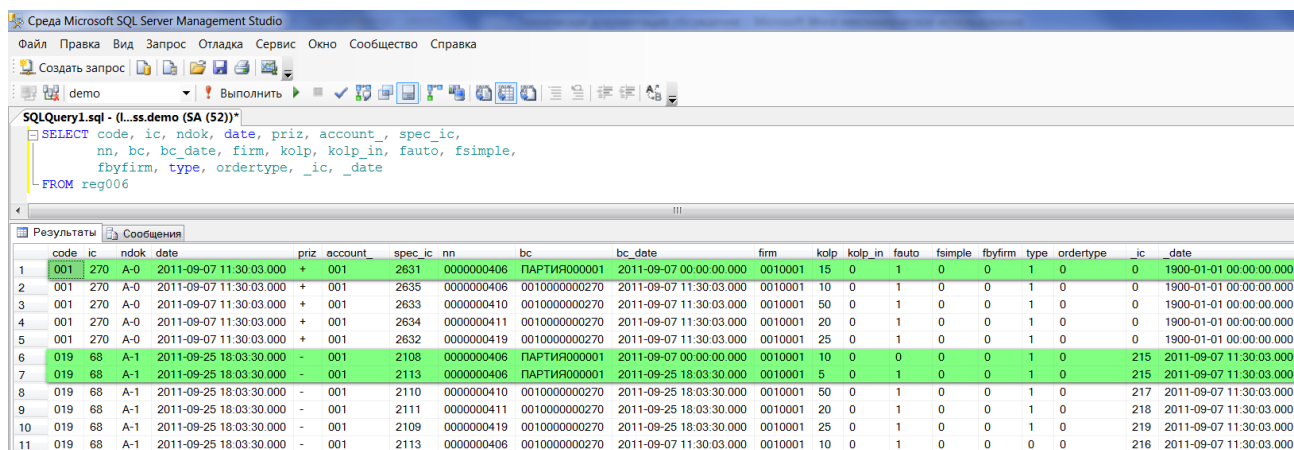
1. Приходную накладную № А-0.
2. Внутреннее перемещение № А-1.

Оба эти документа осуществляют движения номеров партий. Причем, в обоих документах для первой строки указан конкретный номер партии, а для остальных строк система автоматически сгенерировала номера партий. Обратите внимание, что в приходной накладной у нас было указано количество 15 для номера партии ПАРТИЯ000001, а во внутреннем перемещении мы указали значение 10, для этого номера партии.



Рассмотрим совместное влияние этих документов на регистр учета партий. Для этого выполним следующий запрос:

```
SELECT code, ic, ndok, date, priz, account_, spec_ic,
        nn, bc, bc_date, firm, kolp, kolp_in, fauto, fsimple,
        fbyfirm, type, ordertype, _ic, _date
FROM reg006
```



На рисунке отмечена запись прихода номера партии и две записи, соответствующие расходу этого номера партии. Рассмотрим более подробно первую запись результата запроса.

Имя поля	Значение	Комментарий
code	001	Код типа документа Приходная накладная.
ic	270	Идентификатор документа. Равно полю identity_column в таблице spr001
ndok	A-0	Номер документа
date	2011-09-07 11:30:03.000	Дата документа. В стандартной конфигурации все документы, кроме внутреннего перемещения и пересортицы создают записи датой документа. Внутреннее перемещение и пересортица списывают номера партий датой документа, а приходят датой прихода.
priz	+	Добавление номера партии.
account_	001	Код модели учета, по которой произведен учет.
spec_ic	2631	Идентификатор строки документа, на основании которой создана запись.
nn	0000000406	Код ТМЦ, для которой указан номер партии.
bc	ПАРТИЯ000001	Учетный номер партии.
bc_date	2011-09-07 00:00:00.000	Информационное поле с датой номера партии. Это может быть срок годности или любая другая указанная в приходе дата.
firm	0010001	Код организации из документа.
kolp	15	Количество ТМЦ с указанным номером партии.
kolp_in	0	Несписанное количество с указанного номера партии. Поле актуально только для записей, приходящих номера партий.
fauto	1	Признак автоматического распределения номера партии. Признак актуален только для записей списывающих номера партий.
fsimple	0	Признак использования упрощенной схемы списания (см. выше).
fbyfirm	0	Признак ведения учета номеров партий в разрезе организаций.
type	1	Тип прихода / списания (1 - Закуп/Продажа, 2 - Комиссия, 3 - Бонус, 4 - Прочее)
ordertype	0	Необходимый порядок сортировки приходов при списании (0 - без учета типа прихода, 1 - Закуп-Комиссия-Бонус-Прочее, 2-...)
_ic	0	Ссылка на запись в регистре reg006, по которой производилось списание. Обратите внимание на значение этого поля для записей, осуществляющих списание партий.
_date	1900-01-01 00:00:00.000	Дата из списанной записи.

**Примечание.** Из результата выполнения запроса видно на примере ТМЦ с кодом 0000000406, как система распределяет номера партий из разных строки прихода по строкам расходам. Обратите внимание на то, с каких строк (поле \_ic) было произведено списание каждой строки расхода.

## Регистр учета оборотной тары.

Регистр учета оборотной тары предназначен для сбора информации о движении оборотной тары. Оборотная тара отличается от обычного товара тем, что её учет ведется в количественном выражении без учета её стоимости. Учет ведется в разрезе организации, контрагентов и оборотной тары.

В стандартной конфигурации следующие документы могут осуществлять движения оборотной тары:

1. Поступление оборотной тары – приходная накладная.
2. Отгрузка оборотной тары – расходная накладная.
3. Возврат (отгрузка) оборотной тары – возврат поставщику.

Особенностью данного регистра является то, что все записи в него заносятся с признаком (поле *priz*) равным '+'.

Таблица reg007.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). В данном регистре в стандартной конфигурации все записи добавляются с признаком '+'.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
client	char(7)	Контрагент, которому отгруженная / получена тара.
firm	char(7)	Код организации, осуществившей операцию.
nn	char(10)	Код тары из справочника ТМЦ.
sklad	char(3)	Склад, на который поступила или была отгружена тара.
kolp_i	float	Количество принятой тары.
kolp_o	float	Количество отгруженной тары.

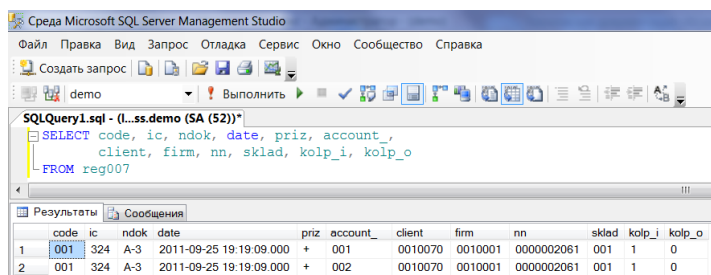
**Примечание.** В стандартной конфигурации учет задолженности по таре ведется в полях *kolp\_i* и *kolp\_o*. Таким образом, для расчета задолженности на указанную дату необходимо сложить все поступления и вычесть сумму всех отгрузок тары. Для получения списка тары с указанием задолженностей по ней необходимо пользоваться хранимой функцией: *fn\_calctare( @tare char(10), @client char(7), @date datetime, @account\_ char(3) );*

### Примеры заполнения регистра учета залоговой тары и использования функции.

Создадим приходную накладную, как указано на картинке. Контрагент может быть выбран любой, склад тоже любой. Обратите внимание на отмеченные поля, установка которых определяет, будет ли документ влиять на регистр учета залоговой тары.

Для просмотра влияния данного документа на регистр выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_,
       client, firm, nn, sklad, kolp_i, kolp_o
FROM reg007
```



Рассмотрим более подробно первую запись результата запроса.

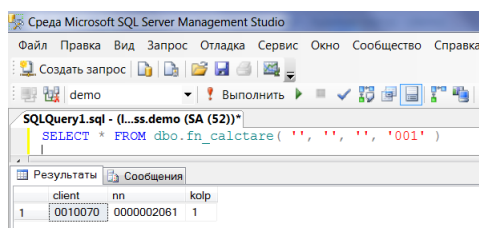
Имя поля	Значение	Комментарий
code	001	Код типа документа Приходная накладная.
ic	324	Идентификатор документа. Равно полю identity_column в таблице spr001
ndok	A-3	Номер документа
date	2011-09-25 19:19:09.000	Дата документа.
priz	+	Добавление соответствующей тары в расчеты.
account_	001	Код модели учета, по которой произведен расчет.
client	0010070	Код контрагента, от которого поступила тара.
firm	0010001	Код организации из документа.
nn	0000002061	Код поступившей тары из справочника ТМЦ
sklad	001	Склад, на котором произведена операция.
kolp_i	1	Поступившее количество
kolp_o	0	Списанное количество.

*Функция fn\_calctare расчет задолженности по таре на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@tare	char(10)	Код тары из справочника ТМЦ. Если не указан, то расчет будет произведен для всех видов тары.
@client	char(7)	Код контрагента, для которого необходимо получить расчет. Если не указан, то будет возвращена информация по всем контрагентам, с которыми были расчеты по таре
@date	datetime	Дата расчета задолженности.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Функция возвращает остаток по той таре и тем контрагентам, по которым было движение в регистре до указанной даты.

```
SELECT * FROM dbo.fn_calctare( '', '', '', '001')
```



### Регистр учета цен реализации.

Регистр учета цен реализации предназначен для сбора информации о движениях товаров в ценах реализации. Также в этом регистре хранится информация об изменении цен реализации. В системе Айтида цены реализации товаров делятся на категории. Количество и параметры категорий цен задается в параметрах системы. Эта информация сохраняется в справочнике категорий цен реализации (*sprccnames*). Категория цены может быть привязана к складу, для определения его учетной цены. В регистре учета цен реализации ведется учет операций в разрезе товаров и категорий цен.

В отличие от других регистров, в регистре учета цен реализации кроме операций прихода с признаком '+' и расхода с признаком '-' используется операция изменения цены. Операция изменения цены называется переоценкой. В этой операции, в качестве количественного показателя используется остаток товара суммарно на всех складах, для которых в качестве учетной цены указана переоцениваемая категория цены. Таким образом, по данным из регистра учета цен реализации можно определить сумму, на которую была произведена переоценка товара.

В регистре учета цен реализации предусмотрена возможность фиксации "акций", приводящих к временному изменению цены реализации. При этом фиксируется дата и время начала акции и дата и время ее окончания. Стандартные функции определения цены товара учитывают возможность наличия акций и сроков их окончания.

Для активации ведения учета цен реализации необходимо в параметрах системы установить признак "Вести учет ТМЦ в ценах реализации".

В стандартной конфигурации следующие документы добавляют записи в регистр учета цен реализации с признаком (поле *priz*) '+':

1. Приходная накладная.
2. Оприходование излишков ТМЦ.
3. Возврат от покупателя.
4. Инвентаризация ТМЦ.
5. Выпуск и комплектация.
6. Внутреннее перемещение.
7. Пересортица ТМЦ.

Следующие документы добавляют записи с признаком (поле *priz*) '-':

1. Расходная накладная.
2. Списание ТМЦ.
3. Инвентаризация ТМЦ.
4. Документ кассовой смены.
5. Выпуск и комплектация.
6. Внутреннее перемещение.
7. Пересортица ТМЦ.
8. Возврат поставщику.

Отдельную операцию переоценки (изменения цены) выполняет документ Переоценка ТМЦ.

Таблица reg008.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-').
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
nn	char(10)	Код ТМЦ
clev	char(3)	Категория переоцениваемой цены
change_cena	int	Признак, если равен 1, то операция изменяет учетную цену (переоценка). Используется тип данных int для того, чтобы можно было создать индекс по этому полю.
f_action	bit	Признак того, что переоценка является акцией
endaction	datetime	Срок окончания акции
set_cena	int	Признак, если равен 1, то операция должна установить цену операции (поле cena) равной текущей учетной цене
kolp	float	Количество товара
cena	float	Цена ТМЦ
cena_	float	Текущая учетная цена
firm	char(10)	Код фирмы, в которой происходит изменение

**Примечание.**

1. В стандартной конфигурации документы влияют на регистр учета цен реализации, если
  - a. установлен признак “Вести учет ТМЦ в ценах реализации” в параметрах системы;
  - b. склад, указанный в документе находится в текущем филиале;
  - c. у склада выбрана учетная цена отличная от Себестоимости;
2. При работе с регистром учета цен реализации производятся проверки:
  - a. если документ не устанавливает новую цену, поле `change_cena = 0`, то проверяется наличие ранее установленной цены;
  - b. при удалении записи, которая устанавливала новую цену (`change_cena = 1`), не допускается возникновения ситуации отсутствия установленной цены для оставшихся в регистре записей;
  - c. если документ устанавливает новую цену, то проверяется отсутствие аналогичного документа с той же датой и временем;
  - d. если добавляется запись, устанавливающая новую цену, то проверяется признак “Разрешить более поздние переоценки” в карточках складов, с изменяемой категорией цены;
  - e. при добавлении/удалении записей “задним числом” автоматически производится пересчет всех более поздних записей для указанного товара, категории цены и модели учета.
3. Для расчета цены товара необходимо использовать функцию:
 


```
fn_calcclev_cena( @nn char(10), @date datetime, @clev char(3), @account_ char(3));
```

**Примеры заполнения регистра учета залоговой тары и использования функции.**

Для примера рассмотрим два документа, которые влияют на данный регистр.

1. Переоценка ТМЦ.
2. Приходная накладная.

Для этого создадим документ переоценка, на основании приходной накладной № А-0, введенной нами ранее, следующим образом:

1. В документе приходная накладная нажмем кнопку Сформировать переоценку 
2. В форме Переоценка по приходу отмечаем все записи и нажимаем кнопку Продолжить.
3. В полученном документе в поле Увеличить на % указываем 10 и расцениваем товар.
4. Проводим полученный документ.



Для того, чтобы посмотреть влияние этих документов на регистр учета цен реализации выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_, nn,
       clev, change_cena, f_action, endaction,
       set_cena, kolp, cena, cena_, firm
FROM reg008 WHERE (code='001' AND ndok='A-0') OR (code='005' AND ndok='A-5')
```

code	ic	ndok	date	priz	account_	nn	clev	change_cena	f_action	endaction	set_cena	kolp	cena	cena_	firm
1	005	333	A-5	2011-09-07 11:30:03.000	+	001	0000000406	001	1	0	0	15	15	15	0010001
2	005	333	A-5	2011-09-07 11:30:03.000	+	001	0000000419	001	1	0	0	12	12	12	0010001
3	005	333	A-5	2011-09-07 11:30:03.000	+	001	0000000410	001	1	0	0	18	18	18	0010001
4	005	333	A-5	2011-09-07 11:30:03.000	+	001	0000000411	001	1	0	0	19	19	19	0010001
5	001	270	A-0	2011-09-07 11:30:03.000	+	001	0000000406	001	0	0	1	15	15	15	0010001
6	001	270	A-0	2011-09-07 11:30:03.000	+	001	0000000419	001	0	0	1	25	12	12	0010001
7	001	270	A-0	2011-09-07 11:30:03.000	+	001	0000000410	001	0	0	1	50	18	18	0010001
8	001	270	A-0	2011-09-07 11:30:03.000	+	001	0000000411	001	0	0	1	20	19	19	0010001
9	001	270	A-0	2011-09-07 11:30:03.000	+	001	0000000406	001	0	0	1	10	15	15	0010001
10	001	270	A-0	2011-09-07 11:30:03.000	+	002	0000000406	001	0	0	1	15	15	15	0010001
11	001	270	A-0	2011-09-07 11:30:03.000	+	002	0000000419	001	0	0	1	25	12	12	0010001
12	001	270	A-0	2011-09-07 11:30:03.000	+	002	0000000410	001	0	0	1	50	18	18	0010001
13	001	270	A-0	2011-09-07 11:30:03.000	+	002	0000000411	001	0	0	1	20	19	19	0010001
14	001	270	A-0	2011-09-07 11:30:03.000	+	002	0000000406	001	0	0	1	10	15	15	0010001

Основное отличие между тем как влияет на данный регистр документ переоценка и все остальные документы, состоит в том, как заполняются поля `change_cena` и `set_cena`. Рассмотрим более подробно первую запись результата запроса.

Имя поля	Значение	Комментарий
code	005	Код типа документа Переоценка ТМЦ.
ic	333	Идентификатор документа. Равно полю <code>identity_column</code> в таблице <code>spr005</code>
ndok	A-5	Номер документа
date	2011-09-07 11:30:03.000	Дата документа.
priz	+	Для документов, которые приходят ТМЦ на склад и для переоценки значение этого поля равно '+'. Значение этого поля равно '+'.
account_	001	Код модели учета, по которой произведен расчет.
nn	0000000406	Код ТМЦ, участвующего в операции.
clev	001	Категория цены.
change_cena	1	Признак того, что операция является переоценкой.
f_action	0	Если 1, то признак того, что переоценка является временной акцией.
endaction	1900-01-01 00:00:00.000	Срок окончания акции
set_cena	0	Если 1, то признак того, что поле <code>сена</code> должно рассчитываться автоматически. В данном случае поле <code>сена</code> не изменяется.
kolp	0	Количество товара, участвующего в операции. Для переоценки рассчитывается автоматически и равно остатку на всех складах с переоцениваемой категорией цены.
cena	15	Цена операции. Для переоценки берется из документа, для остальных операций рассчитывается автоматически.
cena_	15	Цена после операции. Для переоценки равна новой цене. Для остальных операций равна учетной цене.
firm	0010001	Код Организации из документа.

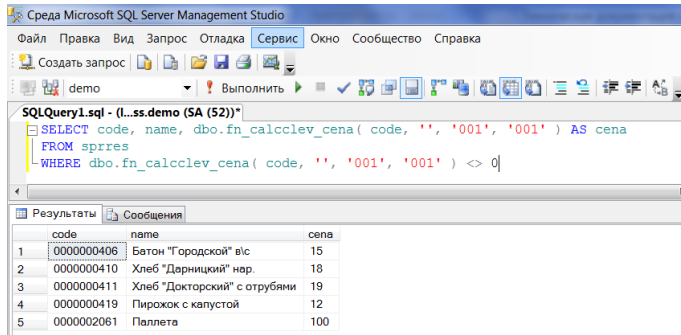
Функция `fn_calcclev_cena` расчет цены на указанную дату.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код тары из справочника ТМЦ, для которого рассчитывается цена.
@date	datetime	Дата расчета цены.
@clev	char(3)	Категория цены, для которой необходимо произвести расчет.

@account_	char(3)	Код модели учета для расчета.
-----------	---------	-------------------------------

Выполним следующий запрос. Функция возвращает цены на указанную дату. Если дата не указана, то возвращается цена на текущую дату. Функция отслеживает наличие акций и срок их окончания.

```
SELECT code, name, dbo.fn_calcclev_cena( code, '', '001', '001' ) AS cena
FROM sprres
WHERE dbo.fn_calcclev_cena( code, '', '001', '001' ) <> 0
```



### Регистр учета серийных номеров.

Регистр учета серийных номеров предназначен сбора информации о движении серийных номеров. *Серийные номера* могут быть указаны для строк документа, содержащих ссылки на товары. В стандартной конфигурации система Айтида ведет учет серийных номеров с учетом указанного их количества. Система не проверяет уникальность серийных номеров, это означает, что ТМЦ с одним серийным номером может быть оприходована более одного раза. Следует отметить, что при списании серийного номера проверки наличия требуемого количества для данного номера не производится.

Следующие документы осуществляют поступление серийных номеров:

1. Приходная накладная.
2. Оприходование излишков ТМЦ.
3. Возврат от покупателя.
4. Внутреннее перемещение.
5. Пересортица ТМЦ.

Следующие документы осуществляют списание серийных номеров:

1. Расходная накладная.
2. Списание ТМЦ.
3. Внутреннее перемещение.
4. Пересортица ТМЦ.
5. Возврат поставщику.

Таблица reg009.

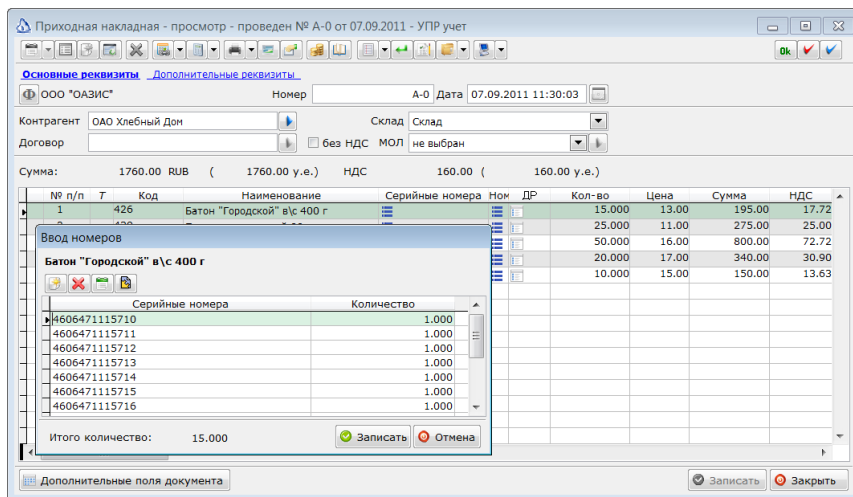
Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-').
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи.
spec_ic	int	Ссылка на спецификацию документа, породившего запись.
nn	char(10)	Код из справочника ТМЦ.
bc	varchar(50)	Серийный номер.
sklad	char(3)	Склад, на котором произошла операция.
kolp	float	Количество списываемое/приходуемое.

#### Примечание.

1. В стандартной конфигурации документы влияют на регистр учета цен реализации, если
  - a. указан не пустой серийный номер;
  - b. склад, на котором производится операция, находится в текущем филиале.
2. Для получения остатка указанного серийного номера указанной ТМЦ необходимо пользоваться функцией:  
`fn_calcbarcodesaldo( @nn char(10), @barcode varchar(50), @date datetime, @account_ char(3) );`

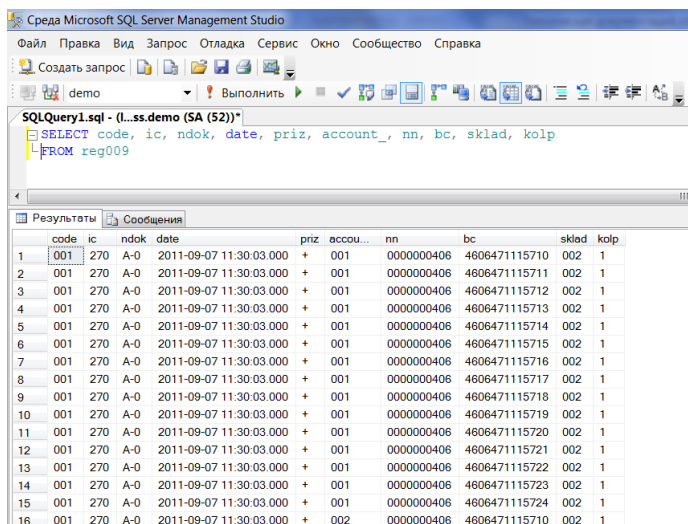
### Примеры заполнения регистра учета серийных номеров и использования функции.

Для примера рассмотрим приходную накладную с номером А-0, введенную ранее. Для первой строки укажем список серийных номеров, как показано на картинке (список условный и не является «настоящими» серийными номерами) и проведем получившийся документ.



Выполним следующий запрос, чтобы изучить, как этот документ повлиял на регистр учета серийных номеров. На рисунке приведена только часть возвращенного результата.

```
SELECT code, ic, ndok, date, priz, account_, nn, bc, sklad, kolp
FROM reg009
```



Рассмотрим более подробно первую запись результата запроса.

Имя поля	Значение	Комментарий
code	001	Код типа документа Приходная накладная.
ic	270	Идентификатор документа. Равно полю identity_column в таблице spr001.
ndok	A-0	Номер документа
date	2011-09-07 11:30:03.000	Дата документа.
priz	+	Для документов, которые приходят ТМЦ на склад значение этого поля равно '+'. Для документов, которые уходят ТМЦ со склада значение этого поля равно '-'.
account_	001	Код модели учета, по которой создана запись.
nn	0000000406	Код ТМЦ, для которой указан серийный номер.
bc	4606471115710	Серийный номер.
sklad	002	Код склада, на котором произведена операция.
kolp	1	Количество ТМЦ с указанным серийным номером.

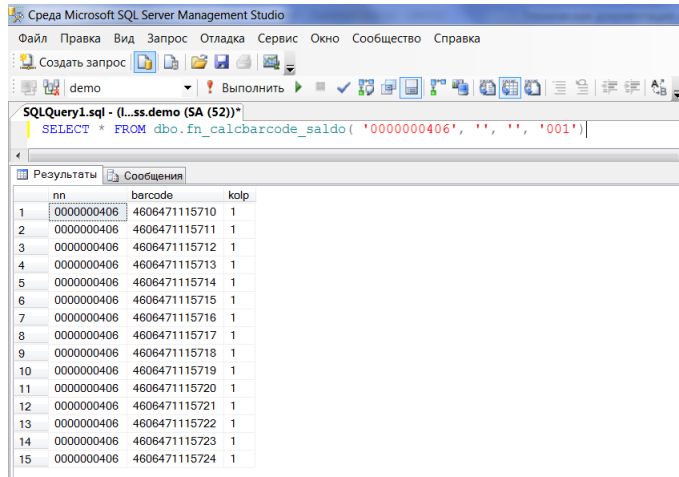
Функция *fn\_calcclev\_cena* расчет цены на указанную дату.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код тары из справочника ТМЦ, для которого рассчитывается цена.

@barcode	varchar(50)	Серийный номер для которого необходимо произвести расчет. Если не указан, то функция вернет список со всеми серийными номерами указанной ТМЦ.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Функция возвращает серийные номера и их количества на указанную дату. Если дата не указана, то возвращаются серийные номера на текущую дату.

```
SELECT * FROM dbo.fn_calbarcode_saldo( '0000000406', '', '', '001')
```



## Регистр учета проводок (журнал хозяйственных операций).

Журнал хозяйственных операций (далее журнал операций) служит для сбора проводок по проводимым хозяйственным операциям. Хозяйственные операции являются основным элементом отражения деятельности в бухгалтерском учете, а также могут использоваться для ведения налогового и управленческого учета. Каждая операция содержит одну или несколько проводок, полностью отражающих в бухгалтерском или управленческом учете хозяйственную операцию. Проводки могут быть сформированы в различных валютах. Данные о хозяйственных операциях предприятия в системе хранятся в журналах операций. Для различных видов учета предполагается наличие разных журналов. Операции могут автоматически формироваться документами, или вводиться вручную. Кроме того, существует специальный механизм закрытия периода, позволяющий формировать проводки по заданному шаблону с использованием указанных пользователем аналитик.

Настройка типовых хозяйственных операций, выполняемых документами системы, производится в справочнике хозяйственных операций. Типовая операция в справочнике представляет собой набор шаблонов проводок, которые будут сформированы при проведении документа. В шаблоне задаются счета дебета и кредита, типы аналитик и другие поля, необходимые для формирования проводки – валюта, сумма, количество, комментарий, условие формирования. Все поля параметризованы и могут рассчитываться в процессе формирования проводки. Проводки в операции могут формироваться по документу в целом, по его спецификации (например, по каждому товару).

Одна операция может содержать проводки, которые с учетом различных заданных условий автоматически формируются в разных журналах операций, по разным планам счетов при проведении документа. Таким образом, однократный ввод данных позволяет отображать информацию для всех принятых на предприятии схем учета.

При грамотно настроенной схеме учета на предприятии, журналы операций содержат наибольшее количество учетных данных. Поэтому, для оптимизации расчетов по этому регистру, система содержит таблицы, предназначенные для хранения промежуточных расчетов, наиболее часто используемых в отчетах.

Влияние документов на этот регистр может быть настроено в каждой конкретной ситуации по-разному. Если на предприятии принято решение вести учет с использованием журналов операций, то все документы производящие какие либо движения, имеющие денежный эквивалент в системе должны отражать эти движения в журнале операций.

К регистру учета проводок относятся следующие таблицы:

1. reg010 – основная таблица, содержащая все сформированные системой проводки.
2. reg\_c010 – Обороты по дням в разрезе счетов, субсчетов и аналитик.
3. reg\_c010\_bs – Свернутые по аналитике обороты по дням в разрезе счетов и субсчетов.
4. reg\_c010\_month – Обороты по месяцам в разрезе счетов, субсчетов и аналитик.
5. reg\_c010\_year – Обороты по годам в разрезе счетов, субсчетов и аналитик.

Актуальность информации во всех таблицах поддерживается автоматически при добавлении, удалении и обновлении данных в таблице reg010. Если в результате, каких либо сбоев произошло рассогласование в данных между таблицами, то необходимо выполнить специальную хранимую процедуру для восстановления согласованности данных. EXECUTE sp\_create\_reg\_c010.

Таблица reg010.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). В данном регистре не используется
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи.
bsk	char(3)	Счет кредита.
ssk	char(3)	Первый субсчет кредита
ssk2	char(3)	Второй субсчет кредита
ssk3	char(3)	Третий субсчет кредита
bsd	char(3)	Счет дебета
ssd	char(3)	Первый субсчет дебета
ssd2	char(3)	Второй субсчет дебета
ssd3	char(3)	Третий субсчет дебета
analit_k	char(13)	Первая аналитика кредита
analit_k2	char(13)	Вторая аналитика кредита
analit_k3	char(13)	Третья аналитика кредита
analit_k4	char(13)	Четвертая аналитика кредита
analit_d	char(13)	Первая аналитика дебета
analit_d2	char(13)	Вторая аналитика дебета
analit_d3	char(13)	Третья аналитика дебета
analit_d4	char(13)	Четвертая аналитика дебета
summa	float	Сумма по проводке
summa_p	float	не используется
cur	char(3)	Валюта проводки
cur_rate	float	Курс конвертации валюты проводки к основной валюте учета системы
proc_	float	Процент к курсу конвертации валюты проводки к основной валюте учета системы
kolp	float	Количество по проводке
note	char(250)	Комментарий / содержание проводки
level_	char(3)	Журнал операций, к которому относится проводка
author	varchar(250)	Пользователь, создавший проводку
status	char(3)	не используется
cross_ic	int	ИД связи, для которой создана проводка
spec_ic	int	ИД строки спецификации, для которой создана проводка
reg004_ic	int	ИД переоценки, для которой создана проводка
summa_sh	float	Предъявленная сумма (например НДС)
f_buybook	bit	Признак того, что проводка была использована в книге покупок

Таблицы reg\_c010, reg\_c010\_month, reg\_c010\_year.

Структура этих таблиц идентична, т.к. данные таблицы содержат однотипную информацию по разным расчетным периодам.

Имя поля	Тип данных	Назначение поля
period	int	Номер периода, за который отражен оборот. Номер периода рассчитывается по следующей формуле. $\text{period} = \text{DATEPART}(\text{year}, \text{ДатаПроводки}) * 65536 + \text{DATEPART}(\text{month}, \text{ДатаПроводки}) * 256 + \text{DATEPART}(\text{day}, \text{ДатаПроводки})$
bs_key	char(73)	Ключ, равный $\text{level\_} + \text{bs} + \text{ss} + \text{ss2} + \text{ss3} + \text{analit} + \text{analit2} + \text{analit3} + \text{analit4} + \text{cur} + \text{account\_}$
level_	char(3)	Код журнала операций
bs	char(3)	Номер счета
ss	char(3)	Номер первого субсчета
ss2	char(3)	Номер второго субсчета
ss3	char(3)	Номер третьего субсчета
analit	char(13)	Первая аналитика
analit2	char(13)	Вторая аналитика

analit3	char(13)	Третья аналитика
analit4	char(13)	Четвертая аналитика
cur	char(3)	Код валюты
account_	char(3)	Код модели учета
debet	float	Сумма дебетового оборота
debet_mc	float	Сумма дебетового оборота в валюте учета
kredit	float	Сумма кредитового оборота
kredit_mc	float	Сумма кредитового оборота в валюте учета
debet_kolp	float	Количественное оборота по дебету
kredit_kolp	float	Количественное оборота по кредиту

**Примечание.** Для ускорения операций поиска в таблице используется поле *bs\_key*, которое содержит в себе значения всех значимых полей. Поэтому, поля *bs*, *ss*, *ss2*, *ss3*, *analit*, *analit2*, *analit3*, *analit4*, *cur*, *account\_* предназначены для внешнего использования и во внутренних расчетах не участвуют.

Таблица *reg\_c010\_bs*.

Имя поля	Тип данных	Назначение поля
period	int	Номер периода, за который отражен оборот. Номер периода рассчитывается по следующей формуле. $period = DATEPART(year, ДатаПроводки) * 65536 + DATEPART(month, ДатаПроводки) * 256 + DATEPART(day, ДатаПроводки)$
bs_key	char(21)	Ключ, равный $level\_ + bs + ss + ss2 + ss3 + cur + account\_$
level_	char(3)	Код журнала операций
bs	char(3)	Номер счета
ss	char(3)	Номер первого субсчета
ss2	char(3)	Номер второго субсчета
ss3	char(3)	Номер третьего субсчета
cur	char(3)	Код валюты
account_	char(3)	Код модели учета
debet	float	Сумма дебетового свернутого оборота
debet_mc	float	Сумма дебетового свернутого оборота в валюте учета
kredit	float	Сумма кредитового свернутого оборота
kredit_mc	float	Сумма кредитового свернутого оборота в валюте учета
debet_kolp	float	Количественное свернутого оборота по дебету
kredit_kolp	float	Количественное свернутого оборота по кредиту

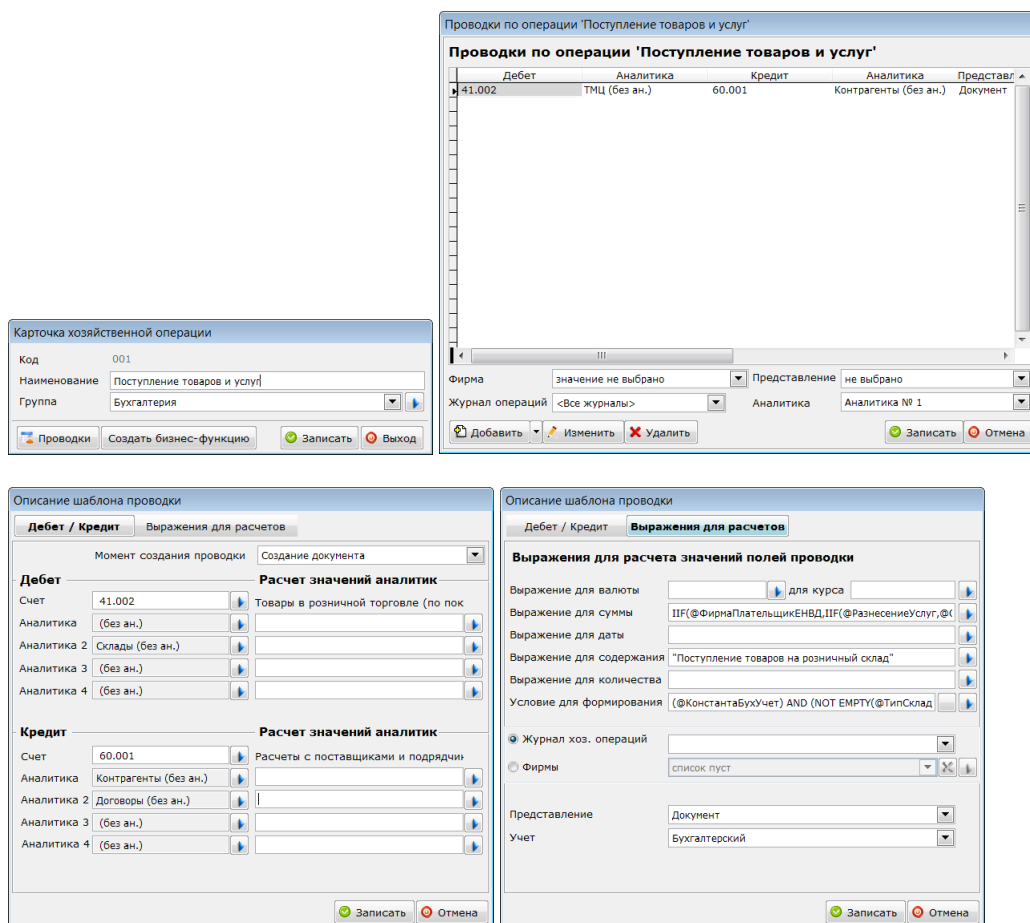
**Примечание.**

- Для ускорения операций поиска в таблице *reg\_c010\_bs* используется поле *bs\_key*, которое содержит в себе значения всех значимых полей. Поэтому, поля *bs*, *ss*, *ss2*, *ss3*, *analit*, *analit2*, *analit3*, *analit4*, *cur*, *account\_* предназначены для внешнего использования и во внутренних расчетах не участвуют.
- Расчет промежуточных оборотов производится автоматически при добавлении, удалении или изменении записей в таблице *reg010*. При изменении записей “задним числом”, система автоматически пересчитывает все данные с более поздними датами.
- В таблицах *reg\_c010*, *reg\_c010\_month*, *reg\_c010\_year* система суммирует обороты по счетам и аналитикам в валюте проводки и в валюте учета. Так как эти таблицы содержат все комбинации счетов и аналитик, то не требуется никакого дополнительного преобразования хранимых сумм.
- В таблице *reg\_c010\_bs* обороты хранятся “свёрнутыми” по аналитике. Это значит, что кредитовый оборот по каждой уникальной комбинации счетов и аналитик (поля *level\_*, *bs*, *ss*, *ss2*, *ss3*, *analit*, *analit2*, *analit3*, *analit4*, *cur*, *account\_*), вычитается из дебетового оборота такой же комбинации счетов и аналитик. Полученный результат, в зависимости от знака, добавляется либо к дебетовому обороту по счету (если сумма получилась положительная), либо к кредитовому (если сумма получилась отрицательная). Таким образом, получается “свёртка” оборота по комбинации аналитик. Данный алгоритм позволяет поддерживать в актуальном состоянии сальдо счетов, на которых может находиться развернутое в общем, но свернутое по аналитике сальдо. Типичным примером такого счета является 60-й или 62-й счета. Общее сальдо этих счетов может быть развернутое, т.е. содержать и суммы и в дебете и в кредите, но внутри все суммы должны быть “свёрнуты” по контрагентам.
- Для получения сальдо счетов на указанную дату необходимо пользоваться функцией:  
*fn\_getbs\_saldo( @filter varchar(8000), @date datetime, level\_ char(3), account\_ char(3), grouping int );*
- Для получения оборотов счетов за указанный период необходимо пользоваться функцией:  
*fn\_getbs\_turnover ( @filter varchar(8000), @date\_min datetime, @date\_max datetime, level\_ char(3), account\_ char(3), grouping int );*



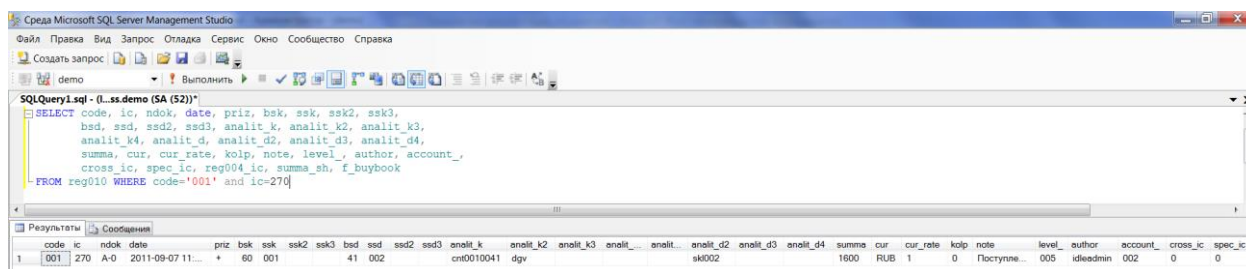
Примеры заполнения регистра проводок.

Для демонстрации данного примера необходимо “забежать” немного вперед и продемонстрировать настройку хозяйственной операции, по которой будут созданы проводки. Для этого необходимо открыть справочник хозяйственных операций и добавить операцию, как указано на картинках.



Созданную операцию присвоим Приходной накладной № А-0 и посмотрим, какие проводки в создались. Для этого выполним запрос:

```
SELECT code, ic, ndok, date, priz, bsk, ssk, ssk2, ssk3,
       bsd, ssd, ssd2, ssd3, analit_k, analit_k2, analit_k3,
       analit_k4, analit_d, analit_d2, analit_d3, analit_d4,
       summa, cur, cur_rate, kolp, note, level_, author, account_,
       cross_ic, spec_ic, reg004_ic, summa_sh, f_buybook
FROM reg010 WHERE code='001' and ic=270
```



Рассмотрим строку результата.

Имя поля	Значение	Комментарий
----------	----------	-------------

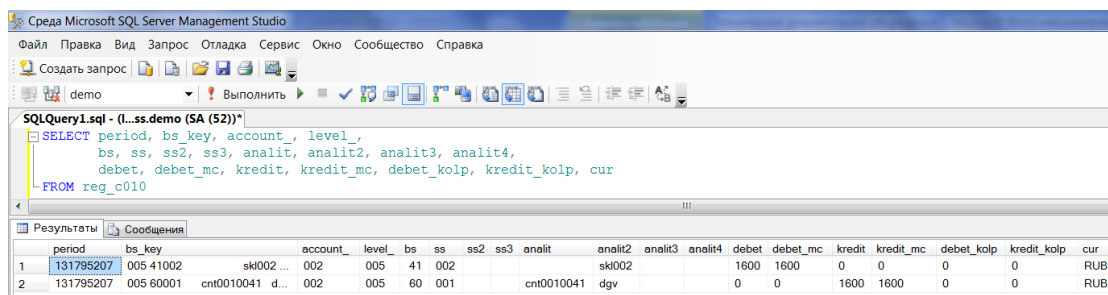
code	001	Код типа документа Приходная накладная.
ic	270	Идентификатор документа. Равно полю identity_column в таблице spr001.
ndok	A-0	Номер документа
date	2011-09-07 11:30:03.000	Дата документа.
priz	+	Для документов, которые приходят ТМЦ на склад значение этого поля равно '+'.
account_	002	Код модели учета, по которой создана запись.
bsk	60	Счет кредита. Был задан в шаблоне проводки.
ssk	001	Субсчет кредита. Был задан в шаблоне проводки.
ssk2		Второй субсчет кредита. Не был задан в шаблоне проводки.
ssk3		Третий субсчет кредита. Не был задан в шаблоне проводки.
bsd	41	Счет дебета. Был задан в шаблоне проводки.
ssd	002	Субсчет дебета. Был задан в шаблоне проводки.
ssd2		Второй субсчет дебета. Не был задан в шаблоне проводки.
ssd3		Третий субсчет дебета. Не был задан в шаблоне проводки.
analit_k	cnt0010041	Первая аналитика кредита. В шаблоне проводки был задан тип аналитики. Система вычислила значение.
analit_k2	dgv	Вторая аналитика кредита. В шаблоне проводки был задан тип аналитики. Система вычислила значение. В данном случае пустое, т.к. поле договор в документе не было заполнено.
analit_k3		Третья аналитика кредита. Не была задана в шаблоне.
analit_k4		Четвертая аналитика кредита. Не была задана в шаблоне.
analit_d		Первая аналитика дебета. Не была задана в шаблоне.
analit_d2	skl002	Вторая аналитика дебета. В шаблоне проводки был задан тип аналитики. Система вычислила значение.
analit_d3		Третья аналитика дебета. Не была задана в шаблоне.
analit_d4		Четвертая аналитика дебета. Не была задана в шаблоне.
summa	1600	Сумма проводки. Система вычислила значение выражения, указанного в шаблоне.
cur	RUB	Валюта проводки, т.к. в шаблоне не было указано выражение для расчета, система использовала валюту документа.
cur_rate	1	Курс конвертации валюты проводки в валюту учета, т.к. в шаблоне не было указано выражение для расчета, система использовала значение из документа.
kolp	0	Количество. В шаблоне не было указано выражение для расчета.
note	Поступление товаров на розничный склад	Комментарий к проводке. Система вычислила значение выражения, указанного в шаблоне.
level_	005	Код журнала операций, в которой создана проводка. Полу заполнено по значению из шаблона.
author	idleadmin	Автор проводки. Заполняется автоматически.
cross_ic	0	Если бы проводка была создана на основании операции сопоставления, то поле бы содержало идентификатор записи связи.
spec_ic	0	Если бы проводка была создана на основании строки многострочной части документа, то поле бы содержало идентификатор записи строки.
reg004_ic	0	Если бы проводка была создана на основании операции переоценки, то поле бы содержало идентификатор записи регистра reg004.
summa_sh	0	Если проводка участвует в книге покупок, то данное поле содержит предъявленную сумму.
f_buybook	0	Признак участия проводки в книге покупок.

Рассмотрим, как заполнились связанные таблицы.

Таблица reg\_c010, reg\_c010\_month, reg\_c010\_year, reg\_c010\_bs.

Рассмотрим заполнение этих таблиц на примере одно из них reg\_c010, т.к. все остальные заполнены аналогично.

```
SELECT period, bs_key, account_, level_,
       bs, ss, ss2, ss3, analit, analit2, analit3, analit4,
       debet, debet_mc, kredit, kredit_mc, debet_kolp, kredit_kolp, cur
FROM reg_c010
```



Как видно из результата запроса исходная проводка была разложена на две записи и для каждой из них был рассчитан свой оборот. Для дебетовой части – оборот по дебету, для кредитовой – по кредиту. Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
period	131795207	Номер периода, для которого рассчитан оборот. В данном случае это 2011 * 65536 + 9 * 256 + 7 = 131795207
bs_key	005 41002 RUB002	Ключ, содержащий всю информацию о счете и аналитика.
account_	002	Код модели учета.
level_	005	Код журнала операций.
bs	41	Счет, для которого произведен расчет оборота.
ss	002	Субсчет счет, для которого произведен расчет оборота.
ss2		Второй субсчет счета. В проводке не был указан.
ss3		Третий субсчет счета. В проводке не был указан.
analit		Первая аналитика счета. В проводке не была указана.
analit2	skl002	Вторая аналитика счета.
analit3		Третья аналитика счета. В проводке не была указана.
analit4		Четвертая аналитика счета. В проводке не была указана.
debet	1600	Сумма дебетового оборота.
debet_mc	1600	Сумма дебетового оборота в валюте учета.
kredit	0	Сумма кредитового оборота.
kredit_mc	0	Сумма кредитового оборота в валюте учета.
debet_kolp	0	Сумма дебетового оборота количества.
kredit_kolp	0	Сумма кредитового оборота количества.
cur	RUB	Валюта расчета.

## Регистр учета расчетов с сотрудниками организаций.

Регистр учета расчетов с сотрудниками организаций предназначен для сбора информации о выданных под отчет и возвращенных (потраченных) суммах денежных средств или их эквивалентов.

В стандартной конфигурации следующие документы начисляют задолженность сотрудника перед организацией:

1. Расходный кассовый ордер.
2. Расход наличных денег.
3. Ценная бумага исходящая.
4. Авансовый отчет.

В стандартной конфигурации следующие документы списывают задолженность сотрудника перед организацией:

1. Товарный чек.
2. Приходный кассовый ордер.
3. Приход наличных денег.
4. Ценная бумага входящая
5. Авансовый отчет.

Документ Авансовый отчет предназначен для фиксации информации о том, на какие цели были потрачены ранее выданные сотруднику денежные средства. Если в этом документе указано, что сотрудник передал денежные средства другому сотруднику, то в стандартной конфигурации с одного сотрудника будет списана задолженность, а другому начислена.

Таблица reg011.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). Признак '+' означает начисление задолженности, признак '-' – списание.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи.
empl	char(7)	Код сотрудника для совершения операции.
summa_po	float	Сумма операции
firm	code(7)	Код организации.

### Примечание.

1. В стандартной конфигурации документы влияют на регистр учета расчетов с сотрудниками, если в качестве получателя (отправителя) указан сотрудник.
2. Для получения информации о задолженности сотрудника перед организацией на указанную дату необходимо использовать функцию:

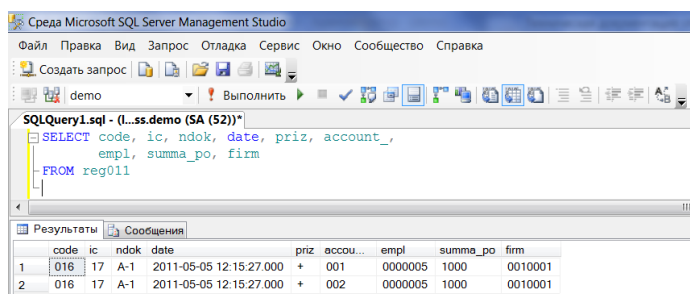
*fn\_calcempl\_saldo( @empl char(7), @firm char(7), @date datetime, @account\_ char(3) );*

## Примеры заполнения регистра расчетов с сотрудниками и выполнения функции.

Для демонстрации примера создадим документ Расходный кассовый ордер, в котором в качестве получателя денежных средств укажем вариант Сотрудник и выберем сотрудника из справочника. Проведем полученный документ.

Для просмотра влияния документа на регистр учета расчетов с сотрудниками организаций выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_,
        empl, summa_po, firm
FROM reg011
```



Рассмотрим первую строку результата.

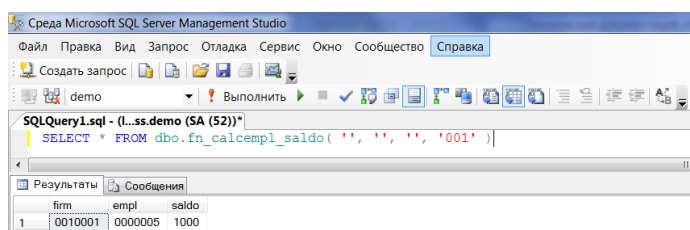
Имя поля	Значение	Комментарий
code	016	Код типа документа Расходный кассовый ордер.
ic	270	Идентификатор документа. Равно полю identity_column в таблице spr016.
ndok	A-0	Номер документа
date	2011-09-07 11:30:03.000	Дата документа.
priz	+	Для документов, которые увеличивают задолженность сотрудника перед организацией значение этого поля равно '+'.
account_	001	Код модели учета, по которой создана запись.
empl	0000005	Код сотрудника из справочника сотрудников организаций.
summa_po	1000	Сумма начисленной задолженности.
firm	0010001	Код организации из документа.

Функция *fn\_calcempl\_saldo* расчет задолженности сотрудника перед организацией на указанную дату.

Имя параметра	Тип параметра	Комментарий
@empl	char(7)	Код сотрудника, для которого необходимо рассчитать задолженность.
@firm	char(7)	Код организации, для которой необходимо рассчитать задолженность.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Функция возвращает код организации, код сотрудника и сумму задолженности на указанную дату. Если дата не указана, то возвращаются задолженности на текущую дату.

```
SELECT * FROM dbo.fn_calcempl_saldo( '', '', '', '001' )
```



## Регистр учета заказов поставщикам.

Система Айтида предусматривает механизмы планирования поставок товаров на склады. Планирование поставок позволяет решать следующие задачи.

1. Прогнозировать состояние складов на требуемую перспективу.
2. Управлять логистикой поставок.
3. Распределять ввод первичной информации между ответственными менеджерами.
4. Автоматизировать контроль реальных поставок.
5. Анализировать отклонения и своевременно предпринимать меры.
6. и т.п.

В стандартной конфигурации планирование поставок осуществляется следующими документами:

1. Счет входящий.
2. Заказ поставщику (не путать с Заказом на поставку)

Факт выполнения плана фиксируют следующие документы:

1. Приходная накладная.
2. Заказ поставщику.

Таблица reg012.

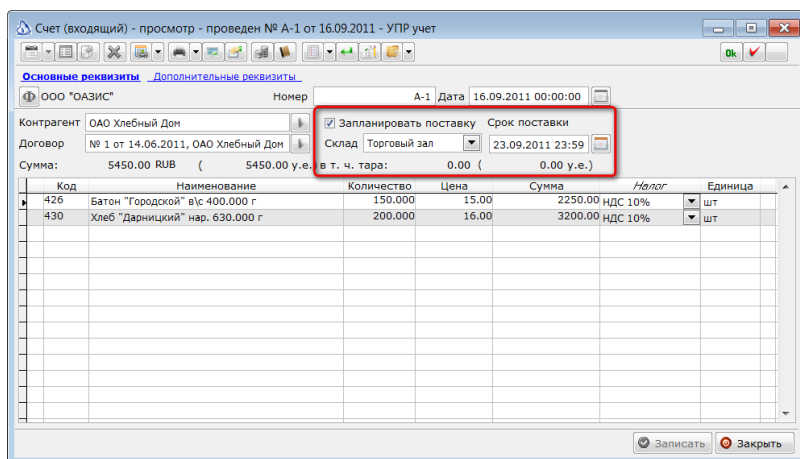
Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак прихода / расхода ('+', '-'). '+' увеличивает план, '-' указывает на факт выполнения.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
client	char(7)	Код контрагента, от которого запланирована поставка ТМЦ.
nn	char(10)	Внутренний код ТМЦ запланированного к поставке
sklad	char(3)	Код склада, на который планируется поставка.
dat_end	datetime	Дата ожидаемой поставки.
kolp	float	Количество, ожидаемое к поставке или поставленное по факту.
rtype	char(1)	не используется. Должно быть равно '1'

### Примечание.

1. В стандартной конфигурации документы влияют на данный регистр, если
  - a. склад, указанный в документе Заказ поставщику находится в текущем филиале;
  - b. склад, указанный в документе Счет входящий находится в текущем филиале и установлен признак Запланировать поставку;
  - c. документ приходная накладная выписан на основании Счета входящего, который запланировал поставку или на основании Заказа поставщику.
2. Для просмотра количества запланированных поставок ТМЦ необходимо пользоваться следующей функцией.  
*fn\_calcdeliveryplan(@nn char(10), @date datetime, @sklad char(3), @account\_ char(3));*
3. Для просмотра количества запланированных поставок ТМЦ с учетом всех подчиненных ТМЦ необходимо пользоваться следующей функцией.  
*fn\_calcdeliveryplansubres (@nn char(10), @date datetime, @sklad char(3), @account\_ char(3));*

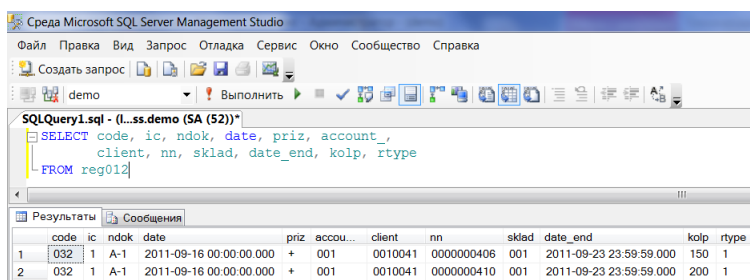
### Примеры заполнения регистра учета заказов поставщикам и использования функций.

Для демонстрации примера заполним документ Счет входящий, как показано на рисунке.



Укажем признак “Запланировать поставку” и укажем срок поставки. Проведем документ. Для просмотра регистра учета заказов поставщикам выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_,
        client, nn, sklad, date_end, kolp, rtype
FROM reg012
```



Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	032	Код типа документа Счет входящий.
ic	1	Идентификатор документа. Равно полю identity_column в таблице spr032.
ndok	A-1	Номер документа
date	2011-09-16 00:00:00.000	Дата документа.
priz	+	Для документов, которые увеличивают план поставки значение этого поля равно '+’.
account_	001	Код модели учета, по которой создана запись.
client	0010041	Код контрагента, от которого запланирована поставка.
nn	0000000406	Код ТМЦ, запланированной к поставке.
sklad	001	Код склада, на который запланирована поставка.
date_end	2011-09-23 23:59:59.000	Срок ожидаемой поставки.
kolp	150	Запланированное к поставке количество ТМЦ.
rtype	1	В стандартной конфигурации значение этого поля всегда равно 1.

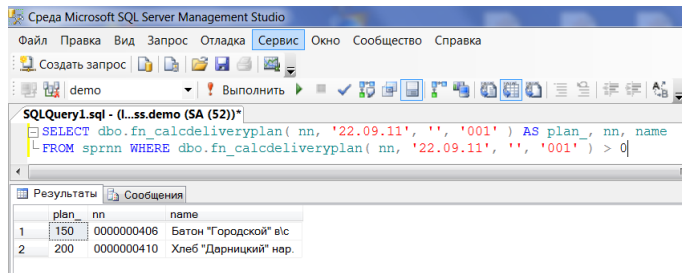
Функция *fn\_calcdeliveryplan* расчет запланированных поставок ТМЦ на указанную дату.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которой необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@sklad	char(7)	Код склада, для которого необходимо рассчитать план поставок.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_calcdeliveryplan( nn, '22.09.11', '', '001' ) AS plan_, nn, name
```

```
FROM sprnn WHERE dbo.fn_calcdeliveryplan( nn, '22.09.11', '', '001' ) > 0
```

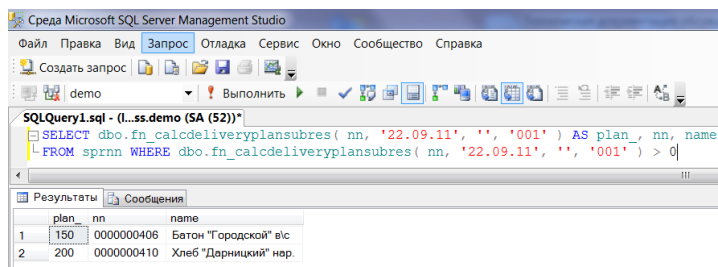


*Функция `fn_calcdeliveryplansubres` расчет запланированных поставок ТМЦ с учетом подчиненных карточек на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код ТМЦ, для которой необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@sklad	char(7)	Код склада, для которого необходимо рассчитать план поставок.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_calcdeliveryplansubres( nn, '22.09.11', '', '001' ) AS plan_, nn, name
FROM sprnn WHERE dbo.fn_calcdeliveryplansubres( nn, '22.09.11', '', '001' ) > 0
```





## Регистр учета бюджетных показателей.

Регистр учета бюджетных показателей занимается сбором информации о плановых и фактических значениях бюджетных показателей. Бюджетный показатель аналогичен балансовому счету, используемому в проводках. У него так же есть три уровня субпоказателей и четыре аналитики. Аналогично проводкам, система Айтида содержит механизмы настройки бизнес-операций, которые, в отличие от проводок, не реализуют принцип двойной записи, а осуществляют одностороннее изменение фактических значений показателей. Плановые значения показателей могут быть загружены в систему при помощи механизма импорта данных из внешней системы бюджетирования.

В тоже время использование плановых показателей не обязательно. Например, типичной реализацией использования данного регистра является построение отчета о движении денежных средств (ДДС). Далеко не всегда существует возможность построить данный отчет, используя только проводки и счета. Очень часто бывает значительно проще создать систему показателей соответствующую структуре ДДС и в денежных документах указывать, к какой статье относится тот или иной документ. После чего, формирование данного отчета становится тривиальной задачей. Кроме того, если в систему загрузить плановые значения по этим же показателям, то будет возможность анализа отклонений фактических значений от плановых.

Как и в случае с хозяйственными операциями, влияние документов на этот регистр может быть настроено в каждой конкретной ситуации по-разному.

Таблица reg013.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак '+' увеличивает план, '-' указывает на факт выполнения.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
bzo	char(3)	Показатель
so	char(3)	Первый субпоказатель
so2	char(3)	Второй субпоказатель
so3	char(3)	Третий субпоказатель
analit	char(13)	Первая аналитика
analit2	char(13)	Вторая аналитика
analit3	char(13)	Третья аналитика
analit4	char(13)	Четвертая аналитика
summa	float	Суммовое выражение показателя
kolp	float	Количественное выражение показателя
note	char(250)	Комментарий к операции
mvz	char(10)	Код центра ответственности
level_	char(3)	Журнал показателей
author	varchar(250)	Пользователь, создавший запись
cross_ic	int	ИД связи, для которой создана запись
spec_ic	int	ИД строки спецификации, для которой создана запись
reg004_ic	int	ИД переоценки, для которой создана запись

**Примечание.** Для получения сальдо показателя на указанный момент времени необходимо использовать функцию:

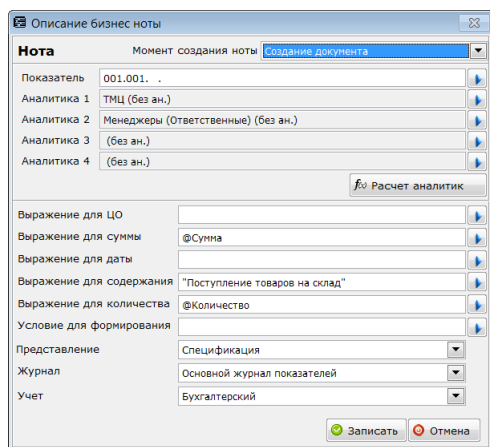
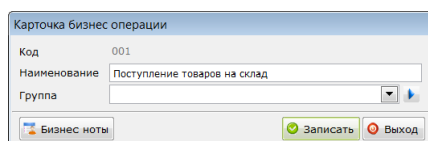
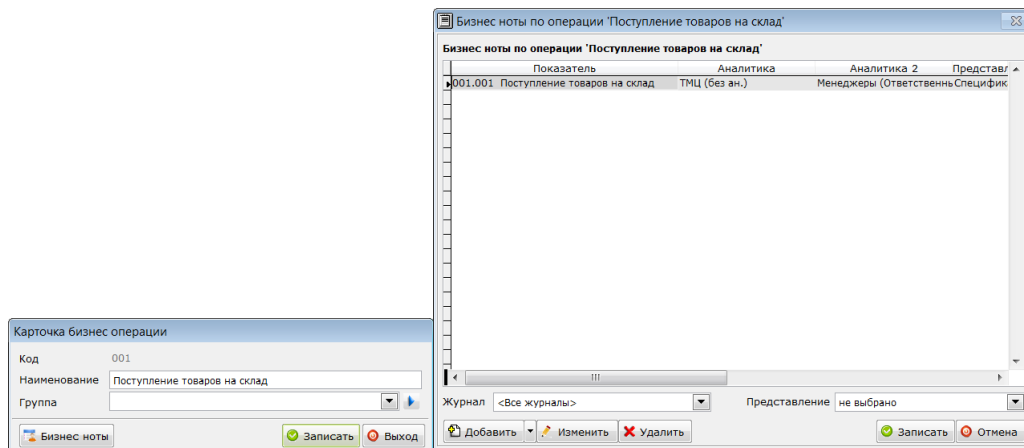
*fn\_getbzo\_saldo( @filter varchar(8000), @date datetime, level\_ char(3), account\_ char(3), grouping int );*

Для получения оборотов по показателю за указанный период необходимо пользоваться функцией:

*fn\_getbzo\_turnover ( @filter varchar(8000), @date\_min datetime, @date\_max datetime, level\_ char(3), account\_ char(3), grouping int );*

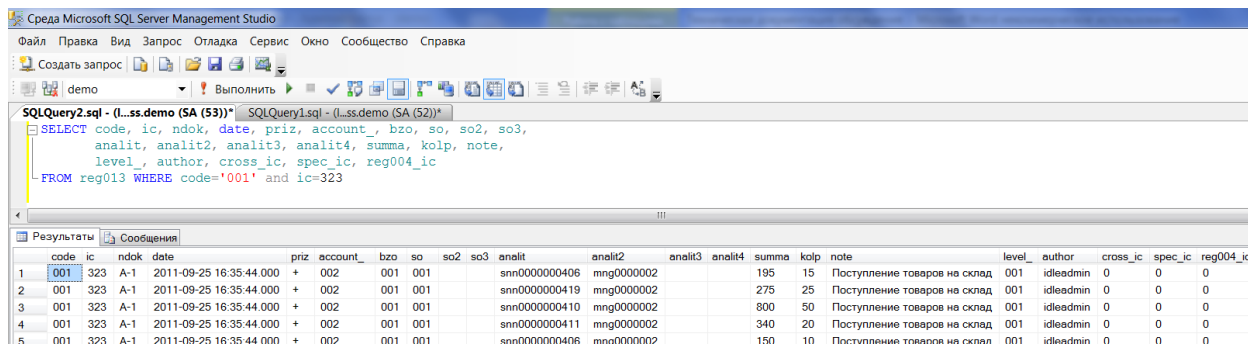
Примеры заполнения регистра учета бюджетных показателей.

Для демонстрации данного примера необходимо настроить бизнес операцию и присвоить ее приходной накладной № А-1. Данная бизнес операция предназначена для учета фактического показателя поступления ТМЦ в разрезе ответственных менеджеров. Для настройки операции необходимо открыть справочник бизнес операций и добавить операцию, как указано на картинках.



Созданную операцию присвоим Приходной накладной № А-1 и посмотрим, какие движения показателей создались. Для этого выполним запрос:

```
SELECT code, ic, ndok, date, priz, account_, bzo, so, so2, so3,
       analit, analit2, analit3, analit4, summa, kolp, note,
       level_, author, cross_ic, spec_ic, reg004_ic
FROM reg013 WHERE code='001' and ic=323
```



Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	001	Код типа документа Приходная накладная.
ic	323	Идентификатор документа. Равно полю identity_column в таблице spr001.
ndok	A-1	Номер документа
date	2011-09-25 16:35:44.000	Дата документа.
priz	+	Для документов, которые приходят ТМЦ на склад значение этого поля равно '+'.
account_	002	Код модели учета, по которой создана запись.
bzo	001	Показатель
so	001	Субкод показателя
so2		Второй субкод показателя. В шаблоне не был указан.
so3		Третий субкод показателя. В шаблоне не был указан.
analit	snn0000000406	Первая аналитика показателя. В шаблоне бизнес – операции был задан тип аналитики. Система вычислила значение.
analit2	mng0000002	Вторая аналитика показателя. В шаблоне бизнес – операции был задан тип аналитики. Система вычислила значение.
analit3		Третья аналитика показателя. В шаблоне бизнес – операции не была задана аналитика.
analit4		Четвертая аналитика показателя. В шаблоне бизнес – операции не была задана аналитика.
summa	195	Сумма. В шаблоне было указано выражение. Система вычислила значение.
kolp	15	Количество. В шаблоне было указано выражение. Система вычислила значение.
note	Поступление товаров на склад	Комментарий. В шаблоне было указано выражение. Система вычислила значение.
level_	001	Код журнала бизнес – операций. В шаблоне было указано конкретное значение. Код модели учета система определяет из параметров выбранного журнала.
author	idleadmin	Автор записи. Заполняется автоматически.
cross_ic	0	Если бы запись была создана на основании операции сопоставления, то поле бы содержало идентификатор записи связи.
spec_ic	0	Если бы запись была создана на основании строки многострочной части документа, то поле бы содержало идентификатор записи строки.
reg004_ic	0	Если бы запись была создана на основании операции переоценки, то поле бы содержало идентификатор записи регистра reg004.

## Универсальный регистр учета.

Универсальный регистр учета используется в случаях, когда необходимо производить учет объектов, для которых не предусмотрены стандартные регистры учета. Данный регистр позволяет учитывать любой объект системы, в том числе объекты, добавленные в процессе настройки системы, используя при этом любую комбинацию параметров учета и любую схему списания. Для настройки универсального регистра учета в системе предусмотрен Справочник правил проведения по универсальному регистру учета. Отчеты по универсальному регистру составляются в генераторе отчетов в произвольной форме, в зависимости от существующих у пользователя настроек.

Как в случае с хозяйственными операциями и показателями, влияние документов на этот регистр может быть настроено в каждой конкретной ситуации по-разному.

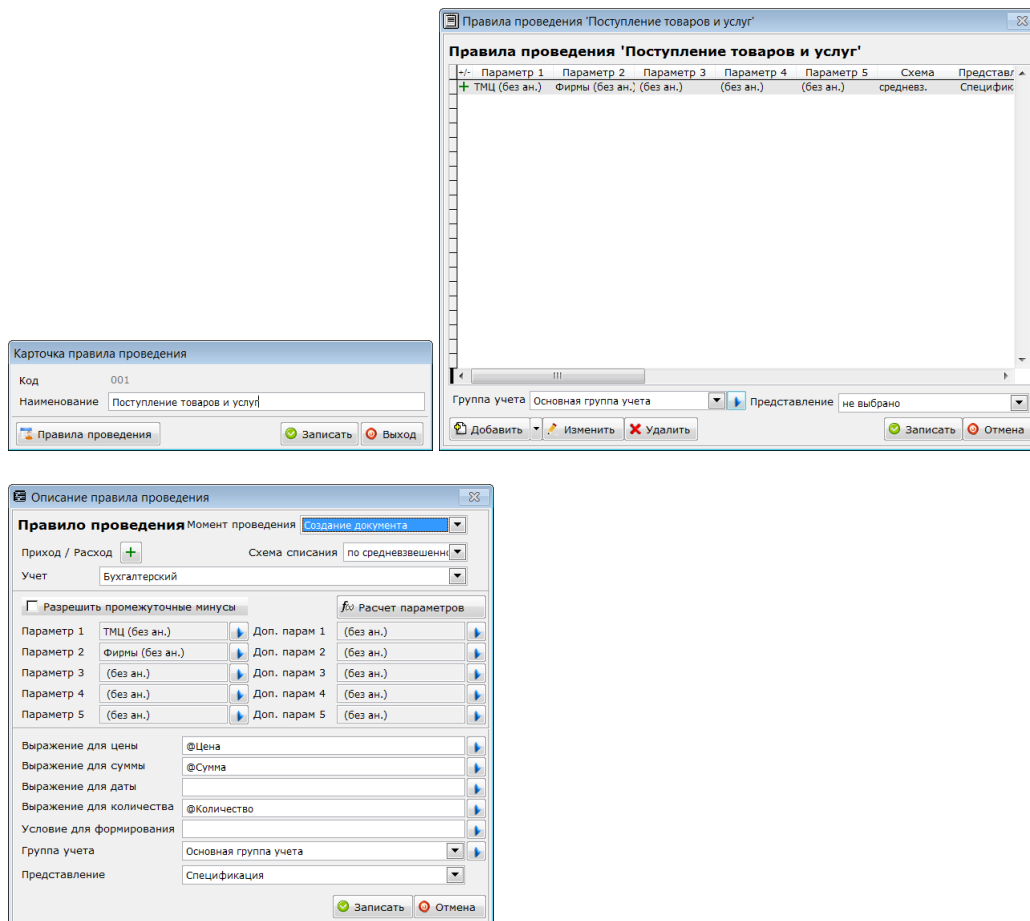
Таблица reg014.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак '+' увеличивает план, '-' указывает на факт выполнения.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи
param1	char(13)	Первый учитываемый параметр
param2	char(13)	Второй учитываемый параметр
param3	char(13)	Третий учитываемый параметр
param4	char(13)	Четвертый учитываемый параметр
param5	char(13)	Пятый учитываемый параметр
level_	char(3)	Группа учета
cena	float	Цена операции
_cena	float	Цена объектов учета до операции
cena_	float	Цена объектов учета после операции
kolp	float	Количество участвующее в операции
_kolp	float	Количество объектов учета до операции
kolp_	float	Количество объектов учета после операции
summa	float	Сумма операции
_summa	float	Сумма остатка объектов учета до операции
summa_	float	Сумма остатка объектов учета после операции
kolp_in	float	Не распределенное количество по приходу. Используется в схемах FIFO и LIFO.
_ic	int	Ссылка на запись, по которой производилось списание. Используется в схемах FIFO и LIFO.
_date	datetime	Дата, по которой производилось списание. Используется в схемах FIFO и LIFO.
param1_ex	char(13)	Первый дополнительный, информационный параметр
param2_ex	char(13)	Второй дополнительный, информационный параметр
param3_ex	char(13)	Третий дополнительный, информационный параметр
param4_ex	char(13)	Четвертый дополнительный, информационный параметр
param5_ex	char(13)	Пятый дополнительный, информационный параметр
type	char(1)	Тип учетной схемы. 'F' - FIFO, 'L' - LIFO, 'S' - средневзвешенная, пусто - никакая.
f_bsubzero	bit	Признак разрешения отрицательных остатков. Используется для никакой схемы списания и для средневзвешенной схемы.
cross_ic	int	Ссылка на связь, породившую данную запись
spec_ic	int	ИД строки спецификации, породившую данную запись
reg004_ic	int	ИД переоценки, для которой создана запись

**Примечание.** Так как данный регистр полностью настраивается под нужды учета, predeterminedных функций для выборки данных не предусмотрено.

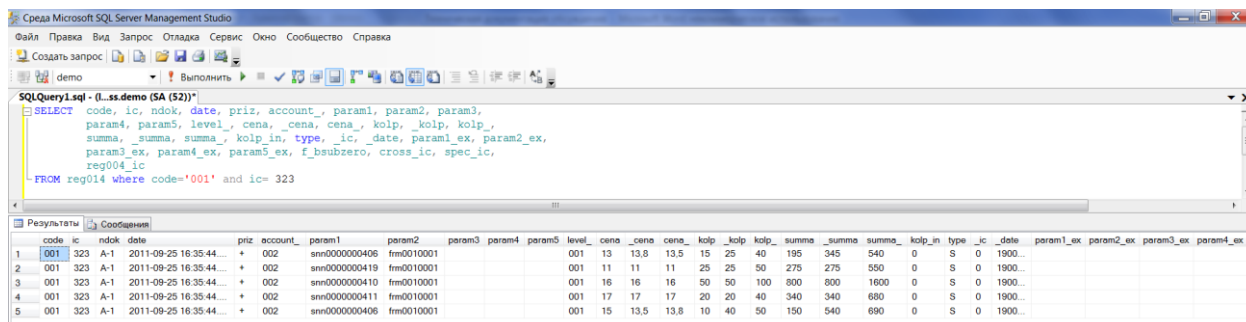
## Примеры заполнения универсального регистра учета.

Для демонстрации данного примера необходимо настроить правила проведения для универсального регистра учета и присвоить их приходной накладной № А-1. Предлагаемое правило предназначено для учета ТМЦ в разрезе организаций по средневзвешенной цене прихода. Для добавления нового правила необходимо открыть справочник правил проведения и добавить новое правило, как указано на картинках.



Созданное правило присвоим Приходной накладной № А-1 и посмотрим, какие движения создались в универсальном регистре. Для этого выполним запрос:

```
SELECT code, ic, ndok, date, priz, account_, param1, param2, param3,
       param4, param5, level_, cena, _cena, cena_, kolp, _kolp, kolp_,
       summa, _summa, summa_, kolp_in, type, _ic, _date, param1_ex, param2_ex,
       param3_ex, param4_ex, param5_ex, f_bsubzero, cross_ic, spec_ic,
       reg004_ic
FROM reg014 where code='001' and ic= 323
```



Рассмотрим первую строку результата.

Имя поля	Значение	Комментарий
code	001	Код типа документа Приходная накладная.
ic	323	Идентификатор документа. Равно полю identity_column в таблице spr001.
ndok	A-1	Номер документа
date	2011-09-25 16:35:44.000	Дата документа.
priz	+	Для документов, которые приходят ТМЦ на склад значение этого поля равно '+’.
account_	002	Код модели учета, по которой создана запись.
param1	snn0000000406	Первый учетный параметр. В данном случае ТМЦ с кодом 0000000406.
param2	frm0010001	Второй учетный параметр. В данном случае Организация с кодом 0010001.
param3		Третий учетный параметр. Пустой, т.к. не был указан в правиле проведения.
param4		Четвертый учетный параметр. Пустой, т.к. не был указан в правиле проведения.
param5		Пятый учетный параметр. Пустой, т.к. не был указан в правиле проведения.
level_	001	Группа учета. По этой группе можно разделять учет однородных параметров по разным схемам учета.
cena	13	Цена операции. В правиле было задано выражение, значение которого было вычислено.
_cena	13,8	Средневзвешенная цена до выполнения операции. В данном случае, перед документом А-1 был проведен документ А-0 (см. выше), в котором была цена 13,8. Значение рассчитано автоматически.
cena_	13,5	Средневзвешенная цена после выполнения операции. Результат взвешенного усреднения. Значение рассчитано автоматически.
kolp	15	Количество операции. В правиле было задано выражение, значение которого было вычислено.
_kolp	25	Количество до операции. Значение рассчитано автоматически.
kolp_	40	Количество после операции. Значение рассчитано автоматически.
summa	195	Сумма операции. В правиле было задано выражение, значение которого было вычислено.
_summa	345	Сумма до операции. Значение рассчитано автоматически.
summa_	540	Сумма после операции. Значение рассчитано автоматически.
kolp_in	0	Поле используется со схемами FIFO или LIFO. В остальных случаях равно 0.
type	S	Тип схемы учета 'S' – средневзвешенная, 'F' – FIFO, 'L' – LIFO, '' – без схемы.
_ic	0	Поле используется со схемами FIFO или LIFO. В остальных случаях равно 0.
_date	1900-01-01 00:00:00.000	Поле используется со схемами FIFO или LIFO. В остальных случаях равно пустой дате.
param1_ex		Дополнительных параметров учета в правиле указано не было, поэтому поля остались пустыми.
param2_ex		
param3_ex		
param4_ex		
param5_ex		
f_bsubzero	0	Признак, разрешающий отрицательные остатки при расчете. Указывается в правиле проведения.
cross_ic	0	Если бы запись была создана на основании операции сопоставления, то поле бы содержало идентификатор записи связи.
spec_ic	0	Если бы запись была создана на основании строки многострочной части документа, то поле бы содержало идентификатор записи строки.
reg004_ic	0	Если бы запись была создана на основании операции переоценки, то поле бы содержало идентификатор записи регистра reg004.

### Регистр учета операций с объектами основных средств и нематериальных активов.

Регистр учета операций с объектами основных средств и нематериальных активов предназначен для накопления информации об учитываемых объектах и для расчета основных показателей учета этих объектов. К таким показателям относятся:

1. Первоначальная стоимость объекта.
2. Балансовая стоимость объекта.
3. Накопленная амортизация.
4. Информация об увеличении или уменьшении стоимости в результате дооборудования или разукрупнения объекта.
5. Информация о местонахождении объекта и МОЛ этого объекта.

В стандартной конфигурации система Айтида не ведет количественный учет объектов, поэтому операции с признаком (поле `priz`) '+' означают увеличение стоимости объекта, а с признаком '-' уменьшение.

В стандартной конфигурации движения с признаком '+' могут создавать следующие документы:

1. Поступление ОС (тип документа 036)
2. Формирование стоимости объекта ОС (тип документа 030)
3. Капитализация объектов ОС (тип документа 031)
4. Движение ОС (тип документа 038)
5. Перемещение стоимости ОС (тип документа 042)
6. Демонтаж ОС (тип документа 057)

В стандартной конфигурации движения с признаком '-' могут создавать следующие документы:

1. Расходная накладная (тип документа 002)
2. Амортизация ОС (тип документа 037)
3. Движение ОС (тип документа 038)
4. Перемещение стоимости ОС (тип документа 042)
5. Демонтаж ОС (тип документа 057)

Одним из основных параметров учета является код операции из "Справочника операций с ОС". Параметры операции определяют, то какие расчеты будут произведены в регистре в момент добавления в него записи. Полный список правил приведен в примечании.

Таблица `reg015`.

Имя поля	Тип данных	Назначение поля
<code>code</code>	<code>char(3)</code>	Код документа, создавшего запись
<code>ic</code>	<code>int</code>	Идентификатор документа, создавшего запись
<code>ndok</code>	<code>char(20)</code>	Номер документа, создавшего запись
<code>date</code>	<code>datetime</code>	Дата операции движения, может не совпадать с датой документа.
<code>priz</code>	<code>char(1)</code>	Признак '+' увеличивает стоимость объекта, '-' уменьшает стоимость объекта.
<code>account_</code>	<code>char(3)</code>	Код модели учета.
<code>transtime</code>	<code>datetime</code>	Дата и время создания записи.
<code>nn</code>	<code>char(10)</code>	Код объекта ОС / нематериального актива.
<code>mol</code>	<code>char(7)</code>	Код МОЛ, за которым числится объект.
<code>mnh</code>	<code>char(10)</code>	Код местонахождение объекта.
<code>amount</code>	<code>float</code>	Перемещаемая стоимость.
<code>kolp</code>	<code>float</code>	Перемещаемое количество. Возможные значения 1 или 0.
<code>oper_</code>	<code>char(10)</code>	Выполняема операция. Код из справочника операций с ОС.

state	char(10)	Состояние объекта после выполнения операции.
f_income	int	Признак того, что операция является оприходованием объекта.
f_operate	int	Признак того, что операция является вводом в эксплуатацию.
f_amort	bit	Признак того, что операция является амортизацией.
f_amortpremium	bit	Признак того, что операция является амортизационной премией.
f_move	int	Признак того, что операция является перемещением объекта.
f_sprinfo	bit	Признак того, что необходимо брать информацию об операции из справочника ОС. Например, начальную стоимость объекта.
period	int	Период операции. Вычисляется по формуле $YEAR(ДатаОперации) * 12 + MONTH(ДатаОперации)$
firm	char(7)	Код организации, выполняющей операцию.

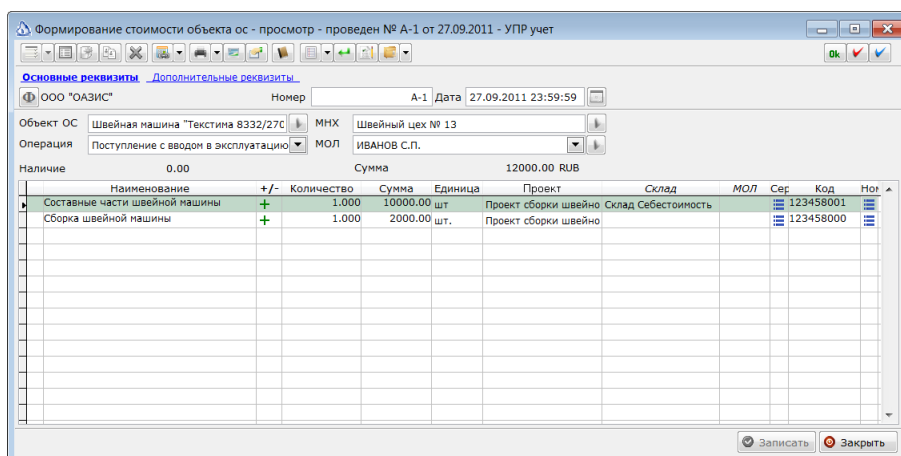
**Примечания.**

1. При добавлении записей в регистр учета операций с объектами основных средств и нематериальных активов проверяются следующие условия:
  - a. отсутствие более поздних амортизаций;
  - b. наличие указанной операции в справочнике операций с ОС;
  - c. при перемещении объекта проверяется его наличие на указанных МНХ и МОЛ;
  - d. при амортизации объекта проверяется отсутствие амортизации в указанном периоде;
  - e. при оприходовании объекта проверяется отсутствие более ранних оприходований этого же объекта;
  - f. при списании объекта проверяется то, что объект был оприходован и еще не был списан;
  - g. при вводе в эксплуатацию проверяется то, что объект еще не был введен в эксплуатацию;
  - h. при изменении стоимости проверяется то, что объект был введен в эксплуатацию;
2. Если в операции указано, что операция должна начислять амортизацию и что необходимо брать информацию из справочника объектов ОС, и в карточке объекта ОС указан начальный износ, то автоматически добавляется запись об амортизации на сумму указанную в карточке объекта. Запись добавляется копированием текущей записи, за исключением полей *priz* – поле устанавливается в значение '-', *amount* – устанавливается в значение начального износа из карточки объекта, *period* – устанавливается в год и месяц исходной операции.
3. Если в операции указано, что операция должна начислять амортизационную премию, и в карточке объекта ОС указан процент амортизационной премии, то автоматически добавляется запись об амортизации на сумму равную указанному проценту от суммы исходно операции. Запись добавляется копированием текущей записи, за исключением полей *priz* – поле устанавливается в значение '-', *amount* – устанавливается в значение начального износа из карточки объекта, *date* – устанавливается в зависимости от того, с какой даты необходимо начислять амортизацию – с дня поступления или на следующий месяц после поступления.
4. Для определения текущего местонахождения объекта необходимо воспользоваться функцией:  
*fn\_osmnh(@code char(10), @date datetime, @account\_ char(3) );*
5. Для определения текущего МОЛ объекта необходимо воспользоваться функцией:  
*fn\_osmol(@code char(10), @date datetime, @account\_ char(3) );*
6. Для определения организации, которая владеет объектом необходимо воспользоваться функцией:  
*fn\_osfirm (@code char(10), @date datetime, @account\_ char(3) );*
7. Для расчета балансовой (остаточной) стоимости объекта необходимо воспользоваться функцией:  
*fn\_osbalancecost (@code char(10), @date datetime, @account\_ char(3) );*
8. Для расчета суммы начисленной амортизации для объекта за период необходимо воспользоваться функцией:  
*fn\_osamortamount (@code char(10), @date\_min datetime, @date\_max datetime, @account\_ char(3) );*
9. Для расчета изменения стоимости объекта за период необходимо воспользоваться функцией:  
*fn\_osamountvariation (@code char(10), @date\_min datetime, @date\_max datetime, @account\_ char(3) );*
10. Для расчета даты принятия объекта к учету необходимо воспользоваться функцией:  
*fn\_osacceptedate (@code char(10), @date datetime, @account\_ char(3) );*
11. Для расчета даты ввода объекта в эксплуатацию необходимо воспользоваться функцией:  
*fn\_osoperatedate (@code char(10), @date datetime, @account\_ char(3) );*
12. Для расчета даты выбытия (реализации, ликвидации и т.п.) объекта необходимо воспользоваться функцией:  
*fn\_osretirementdate (@code char(10), @date datetime, @account\_ char(3) );*



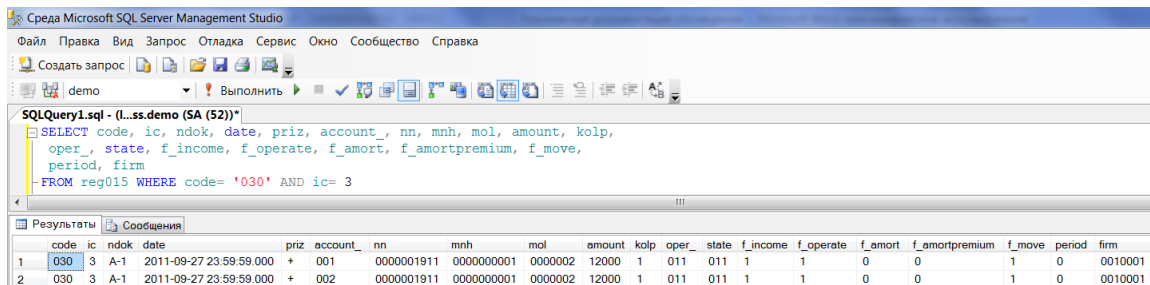
Примеры заполнения регистра учета операций с объектами основных средств и нематериальных активов, а также примеры использования функций.

Для демонстрации примера заведем в систему документ Формирование стоимости ОС следующим образом. Предварительно должны быть оприходованы соответствующие составляющие на склад.



Для просмотра влияния этого документа на учет операций с объектами основных средств и нематериальных активов выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_, nn, mnh, mol, amount, kolp,
       oper_, state, f_income, f_operate, f_amort, f_amortpremium, f_move,
       period, firm
FROM reg015 WHERE code= '030' AND ic= 3
```



Рассмотрим первую запись результата.

Имя поля	Значение	Комментарий
code	030	Код типа документа Формирование стоимости ОС.
ic	3	Идентификатор документа. Равно полю identity_column в таблице spr030.
ndok	A-1	Номер документа
date	2011-09-27 23:59:59.000	Дата документа.
priz	+	Для документов, которые увеличивают стоимость объекта значение этого поля равно '+'. Для документов, которые уменьшают стоимость объекта значение этого поля равно '-'.
account_	002	Код модели учета, по которой создана запись.
nn	0000001911	Код объекта ОС, с которым совершена операция.
mnh	0000000001	Код местонахождения объекта ОС. Имеет смысл, если операция подразумевает перемещение объекта, т.е. f_move = 1.
mol	0000002	Код МОЛ из справочника сотрудников. Имеет смысл, если операция подразумевает перемещение объекта, т.е. f_move = 1.
amount	12000	Сумма операции.
kolp	1	Количество. В стандартной конфигурации всегда 1.
oper_	011	Код выполняемой операции с объектом из справочника операций с ОС.
state	011	Код нового состояния объекта ОС. Устанавливается, если операция

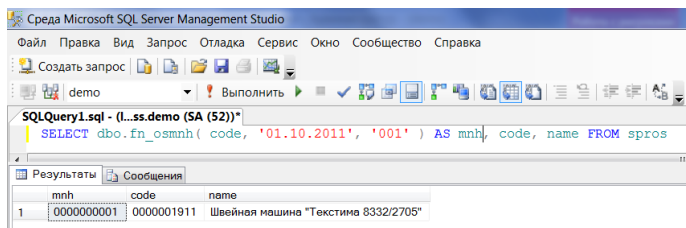
		подразумевает изменение состояния.
f_income	1	Признак поступления (оприходования) объекта. Операция с таким признаком должна быть одна на все время «жизни» объекта. Второй раз объект поступить не может. Перед новым поступлением объект должен быть списан с признаком f_income= -1.
f_operate	1	Признак операции ввода в эксплуатацию. Объект не может быть дважды введен в эксплуатацию. Перед повторным вводом, он должен быть выведен из эксплуатации с признаком f_operate= -1.
f_amort	0	Признак того, что операция является операцией амортизации. В одном периоде объект не может иметь две операции с признаком амортизации.
f_amortpremium	0	Признак операции по начислению амортизационной премии.
f_move	1	Признак операции перемещения объекта между МНХ и МОЛ.
period	0	Код периода амортизации. Имеет смысл только для записей с признаком f_amort = 1.
firm	0010001	Код организации из документа.

*Функция fn\_osmnh расчет местонахождения объекта на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@code	char(10)	Код объекта, для которого необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_osmnh( code, '01.10.2011', '001' ) AS mnh, code, name FROM spros
```

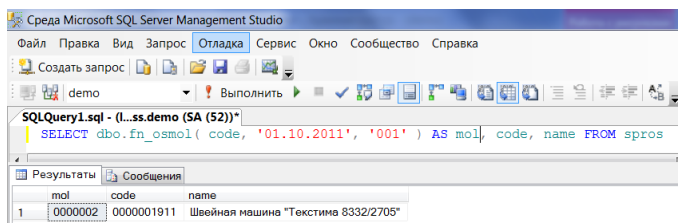


*Функция fn\_osmol расчет МОЛ объекта на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@code	char(10)	Код объекта, для которого необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_osmol( code, '01.10.2011', '001' ) AS mol, code, name FROM spros
```

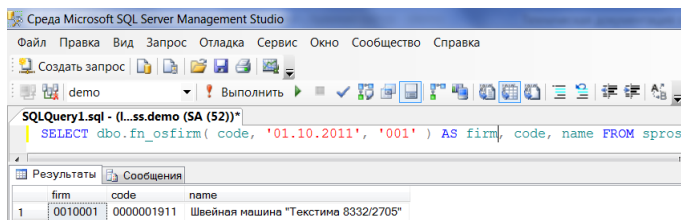


*Функция fn\_osfirm расчет организации, в которой числится объект на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@code	char(10)	Код объекта, для которого необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_osfirm( code, '01.10.2011', '001' ) AS firm, code, name FROM spros
```

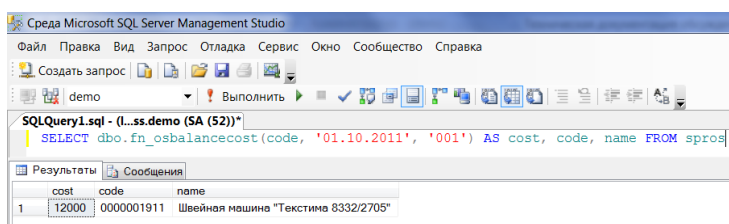


Функция *fn\_osbalancecost* расчет остаточной стоимости объекта на указанную дату.

Имя параметра	Тип параметра	Комментарий
@code	char(10)	Код объекта, для которого необходимо произвести расчет.
@date	datetime	Дата, на которую необходимо произвести расчет.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Если дата не указана, то расчет производится на текущую дату.

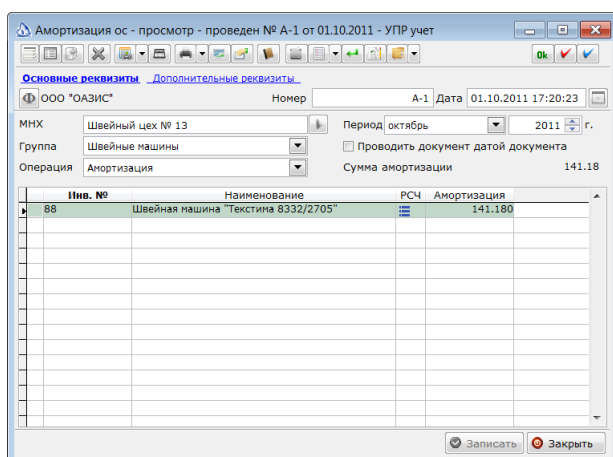
```
SELECT dbo.fn_osbalancecost (code, '01.10.2011', '001') AS cost, code, name FROM spros
```



Функция *fn\_osamortamount* расчет накопленной амортизации объекта за указанный период.

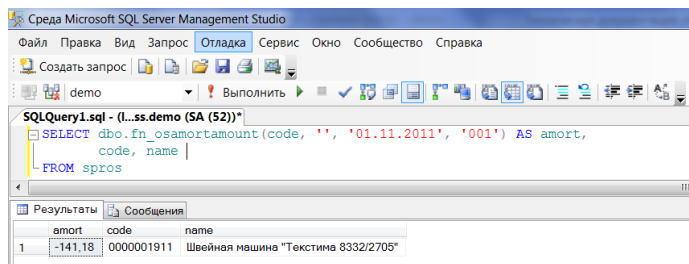
Имя параметра	Тип параметра	Комментарий
@code	char(10)	Код объекта, для которого необходимо произвести расчет.
@date_min	datetime	Дата начала периода расчета.
@date_max	datetime	Дата окончания периода расчета.
@account_	char(3)	Код модели учета для расчета.

Перед выполнением запроса необходимо создать документ, начисляющий амортизацию, как на картинке. Следует иметь в виду, что без указания признака “Проводить документ датой документа”, документ Амортизация ОС создает движения в регистрах последним днем периода амортизации.



Выполним следующий запрос. Если даты не указаны, то расчет производится с самой первой операции по текущую дату.

```
SELECT dbo.fn_osamortamount (code, '', '01.11.2011', '001') AS amort,
       code, name
FROM spros
```



## Регистр учета операций с ценными бумагами.

Регистр учета операций с ценными бумагами предназначен для сбора информации о движении ценных бумаг в организации. В качестве ценной бумагой может быть любой объект имеющий серию, номер, эмитента и т.п. реквизиты. Например, векселя, акции, долговые расписки и т.д. Ценная бумага может быть куплена у контрагента или сотрудника и может быть реализована контрагенту или сотруднику.

В стандартной конфигурации два документа могут осуществлять движения в данном регистре:

1. Ценная бумага входящая – оформляет поступление ценной бумаги на баланс предприятия.
2. Ценная бумага исходящая – оформляет реализацию ценной бумаги.

Таблица reg017.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак '+' приходит ценную бумагу, '-' списывает ценную бумагу.
account	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи.
deposit	char(3)	Код депозитария
bill_series	char(20)	Серия ценной бумаги
bill_number	char(20)	Номер ценной бумаги
date_em	datetime	Дата выпуска (эмиссии) ценной бумаги
emitter	char(7)	Код эмитента из справочника контрагентов
nominal	float	Номинал ценной бумаги
proc_	float	Процент дисконта по ценной бумаге
cur	char(3)	Валюта номинала ценной бумаги
maturity	datetime	Срок погашения ценной бумаги
client	char(7)	Код контрагента, которому продана или у которого куплена ценная бумага
empl	char(7)	Код сотрудника, которому продана или у которого куплена ценная бумага
firm	char(7)	Код организации, проводящий операцию
summa	float	Сумма операции

**Примечание.** В стандартной конфигурации, для того чтобы документ влиял на регистр учета операций с ценными бумагами необходимо заполнить серию, номер ценной бумаги и проверить, что используемый депозитарий находится в текущем филиале.

## Примеры заполнения регистра учета ценных бумаг.

Для демонстрации примера введем в систему документ Ценная бумага входящая и проведем его.

Для просмотра влияния данного документа на регистр учета ценных бумаг выполним следующий запрос:

```
SELECT code, ic, ndok, date, priz, account_, deposit, summa, bill_series,
       bill_number, emitter, client, empl, nominal, proc_, date_em, maturity,
       firm
FROM reg017 WHERE code='039' AND ic= 1
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
SELECT code, ic, ndok, date, priz, account_, deposit, summa,
       bill_series, bill_number, emitter, client, empl, nominal,
       proc_, date_em, maturity, firm
FROM reg017 WHERE code='039' AND ic= 1
```

The Results pane shows the following data:

code	ic	ndok	date	priz	account_	deposit	summa	bill_series	bill_number	emitter	client	empl	nominal	proc_	date_em	maturity	firm
039	1	A-1	2011-09-27 18:00:27.000	+	001	001	45000000	AA-01	123456	0010041	0010041		50000000	10	2011-09-01 00:00:00.000	2011-12-31 00:00:00.000	0010001
039	1	A-1	2011-09-27 18:00:27.000	+	002	001	45000000	AA-01	123456	0010041	0010041		50000000	10	2011-09-01 00:00:00.000	2011-12-31 00:00:00.000	0010001

Имя поля	Значение	Комментарий
code	030	Код типа документа Формирование стоимости ОС.
ic	3	Идентификатор документа. Равно полю identity_column в таблице spr030.
ndok	A-1	Номер документа
date	2011-09-27 23:59:59.000	Дата документа.
priz	+	Для документов, которые увеличивают стоимость объекта значение этого поля равно '+'.
account_	002	Код модели учета, по которой создана запись.
deposit	001	Код депозитария, в котором размещена ценная бумага.
summa	45 000 000	Сумма покупки ценной бумаги.
bill_series	AA-01	Серия ценной бумаги.
bill_number	123456	Номер ценной бумаги.
emitter	0010041	Код контрагента – эмитента ценной бумаги.
client	0010041	Код контрагента – продавца ценной бумаги.
empl		Если покупка была от сотрудника, то был бы код сотрудника.
nominal	50 000 000	Номинал ценной бумаги.
proc_	10	Процент дисконта.
date_em	2011-09-01 00:00:00.000	Дата эмиссии ценной бумаги.
maturity	2011-12-31 00:00:00.000	Срок погашения ценной бумаги.
firm	0010001	Фирма из документа.

## Регистр учета калькуляций и состава объектов ОС.

Регистр учета калькуляций и состава объектов ОС используется для сбора двух видов информации:

1. Состав блюд, указанный в документах Калькуляция.
2. При учете состава объектов ОС на этапах формирования объекта, изменения его состава и выбытия объекта.

В стандартной конфигурации состав блюд заносит в данный регистр один документ – Калькуляция. Информация заносится с признаком '+'. Операций с признаком '-' не предусмотрено. Калькуляция может быть отменена только при наличии более поздней калькуляции или калькуляции с другими параметрами, например сезонной калькуляции.

При учете состава объектов ОС в стандартной конфигурации следующие документы добавляют ресурсы (ТМЦ или работы/услуги) в состав объекта или к проекту, до формирования объекта:

Документ	Тип документа	Комментарий
Приходная накладная	001	добавляет ресурсы к проекту.
Внутреннее перемещение	019	добавляет ресурсы к проекту.
Формирование стоимости объекта ОС	030	добавляет ресурсы в состав объекта.
Капитализация объектов ОС	031	добавляет ресурсы в состав объекта.
Перемещение стоимости ОС	042	добавляет ресурсы в состав объекта.
Акт сдачи/приемки услуг входящий	049	добавляет ресурсы к проекту.
Демонтаж ОС	057	добавляет ресурсы к проекту.

Следующие документы изымают ТМЦ из состава объекта или из проекта, до формирования объекта:

Документ	Тип документа	Комментарий
Расходная накладная	001	изымает ресурсы из проекта.
Списание ТМЦ	019	изымает ресурсы из проекта.
Внутреннее перемещение	030	изымает ресурсы из проекта.
Формирование стоимости объекта ОС	031	изымает ресурсы из проекта.
Перемещение стоимости ОС	042	изымает ресурсы из состава объекта.
Демонтаж ОС	057	изымает ресурсы из состава объекта.

Таблица reg019.

Имя поля	Тип данных	Назначение поля
code	char(3)	Код документа, создавшего запись
ic	int	Идентификатор документа, создавшего запись
ndok	char(20)	Номер документа, создавшего запись
date	datetime	Дата операции движения, может не совпадать с датой документа.
priz	char(1)	Признак '+' приходит ценную бумагу, '-' списывает ценную бумагу.
account_	char(3)	Код модели учета.
transtime	datetime	Дата и время создания записи.
r_code	char(10)	Перемещаемый ресурс. Ресурс может быть перемещен на проект, в состав объекта ОС или использоваться как ингредиент в калькуляции.
nn	char(10)	Объект ОС, в состав или из состава которого перемещается ресурс. Для калькуляций не используется.
project	char(10)	Проект, в который или из которого перемещается ресурс. Для калькуляций не используется.
kolp	float	Перемещаемое / используемое в калькуляции количество ресурса.
amount	float	Перемещаемая сумма. Для калькуляций не используется.
client	char(7)	Контрагент из документа. Для калькуляций не используется.
firm	char(7)	Код организации, выполняющей операцию.

text_	varchar(250)	Текст строки из документа движения. Для калькуляций не используется.
<b>ПОЛЯ ДЛЯ УЧЕТА КАЛЬКУЛЯЦИЙ</b>		
kolp_netto	float	Количество (нетто) списываемых ингредиентов.
prod_out	float	Выход ингредиента после обработки.
ed	varchar(15)	Единица измерения готового блюда, для которой составлена калькуляция.
koef_e	float	Количество в единицах измерения готового блюда, для которого создана калькуляция.
enabled_type	int	Тип срока действия калькуляции (0- бессрочная, 1- сезонная, 2- период)
season	char(10)	Код сезона из справочника сезонов, для которого действует калькуляция
startdt	datetime	Дата начала действия калькуляции, для типа 2 – период.
finishdt	datetime	Дата окончания действия калькуляции, для типа 2 – период.
f_make	bit	Если ИСТИНА - сборка, ИНАЧЕ - разборка
incena_koef	float	Для калькуляции разборки. Коэффициент от себестоимости разбираемого блюда, для расчета себестоимости приходуемого ингредиента.
incena_type	int	Для калькуляции разборки. Тип расчета себестоимости для приходуемого ингредиента. 0 - приходовать по текущей себестоимости, 1 - использовать коэффициент (incena_koef), 2 - ручной ввод себестоимости.
main_incena_type	int	Для калькуляции сборки. Тип расчета себестоимости для готового блюда. 0 – приходовать по текущей себестоимости, 1 - суммировать по составу, 2 - ручной ввод себестоимости.
prod_output	varchar(250)	Выход готового блюда

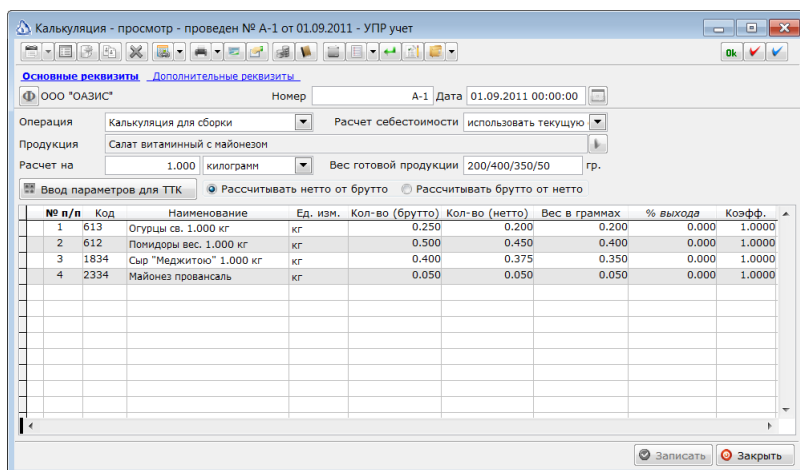
**Примечание.**

1. В стандартной конфигурации документы влияют на регистр учета калькуляций и состава объектов ОС, если:
  - a. документ является калькуляцией и в шапке документа выбрано блюдо, в строке выбран ингредиент (ТМЦ или услуга), входящие в состав блюда.
  - b. документ добавляет ресурс к проекту и в строке документа выбран ресурс (ТМЦ или услуга), указан проект и склад, выбранный в документе, находится в текущем филиале.
  - c. документ изымает ресурс из проекта и в строке документа выбран ресурс (ТМЦ или услуга), указан проект и склад, выбранный в документе, находится в текущем филиале.
  - d. документ добавляет ресурс с состав объекта ОС и выбран добавляемый ресурс и указан объект, в который добавляется ресурс.
  - e. документ изымает ресурс из состава объекта ОС и выбран изымаемый ресурс и указан объект, из которого изымается ресурс.
2. Для получения информации о действующей калькуляции необходимо вызвать хранимую функцию:  
*fn\_getcalc(@nn char(10), @date datetime, @account\_ char(3), @f\_make int );*
3. Для формирования состава блюда на указанную дату необходимо вызвать хранимую процедуру:  
*sp\_assemblageproduct (@nn char(10), @date datetime, @sklad char(3), @account\_ char(3), @f\_make int, @mainkolp float, @f\_virtualcalc int= 0, @f\_notCreateSaldo int= 0)*
4. Для формирования состава объекта ОС на указанную дату необходимо вызвать хранимую функцию:  
*fn\_osstructure( @os char(10), @date datetime, @account\_ char(3));*
5. Для формирования состава проекта на указанную дату необходимо вызвать хранимую функцию:  
*fn\_projectstructure( @project char(10), @date datetime, @account\_ char(3));*

**Примеры заполнения регистра учета калькуляций и состава объектов ОС.**

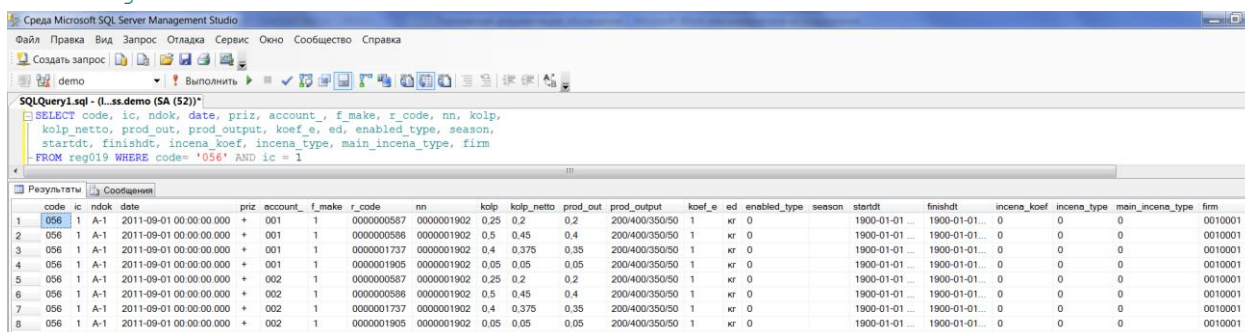
Для демонстрации примера создадим документ Калькуляция, как представлено на картинке и проведем его.





Для просмотра влияния этого документа на регистр учета калькуляций и состава объектов ОС выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_, f_make, r_code, nn, kolp,
       kolp_netto, prod_out, prod_output, koef_e, ed, enabled_type, season,
       startdt, finishdt, incena_koef, incena_type, main_incena_type, firm
FROM reg019 WHERE code= '056' AND ic = 1
```



Рассмотрим первую строку результата запроса.

Имя поля	Значение	Комментарий
code	056	Код типа документа Калькуляция.
ic	1	Идентификатор документа. Равно полю identity_column в таблице spr056.
ndok	A-1	Номер документа
date	2011-09-01 00:00:00.000	Дата документа.
priz	+	В стандартной конфигурации для калькуляций всегда равно '+'.
account_	002	Код модели учета, по которой создана запись.
f_make	1	Признак того, что это калькуляция сборки.
r_code	0000000587	Код ингредиента из справочника ТМЦ.
nn	0000001902	Код блюда из справочника ТМЦ.
kolp	0,25	Количество (брутто) ингредиента в составе. Все расчеты требуемого количества ингредиента строятся по этому полю.
kolp_netto	0,2	Количество (брутто) ингредиента в составе.
prod_out	0,2	Выход ингредиента после переработки. Используется для печати ТТК.
prod_output	200/400/350/50	Общий выход блюда. Используется для печати ТТК.
koef_e	1	Калькуляция на 1 штуку готового блюда.
ed	кг	Единица измерения готового блюда
enabled_type	0	Бессрочная калькуляция.
season		Сезона нет, т.к. бессрочная калькуляция.
startdt	1900-01-01 00:00:00.000	Даты начала и конца действия нет, т.к. бессрочная калькуляция.
finishdt	1900-01-01 00:00:00.000	
incena_koef	0	Это поле используется только для калькуляции "разборка".
incena_type	0	Это поле используется только для калькуляции "разборка".
main_incena_type	0	Это поле используется только для калькуляции "разборка".

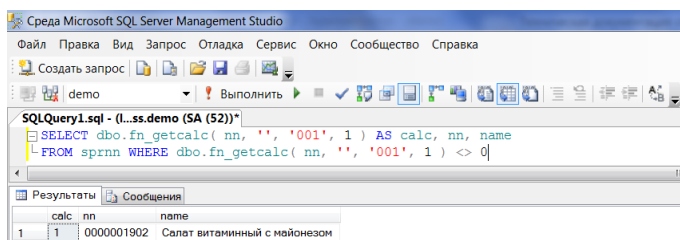
firm	0010001	Код организации из документа.
------	---------	-------------------------------

Функция *fn\_getcalc* вычисление идентификатора активной калькуляции на указанную дату.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код блюда из справочника ТМЦ, для которого производится расчет.
@date	datetime	Дата расчета.
@account_	char(3)	Код модели учета для расчета.
@f_make	int	1 – если требуется калькуляция “сборка”, иначе – 0

Выполним следующий запрос. Он вернет все ТМЦ, для которых определены калькуляции на текущий момент. Если дата не указана, то расчет производится на текущую дату.

```
SELECT dbo.fn_getcalc( nn, '', '001', 1 ) AS calc, nn, name
FROM sprnn WHERE dbo.fn_getcalc( nn, '', '001', 1 ) <> 0
```



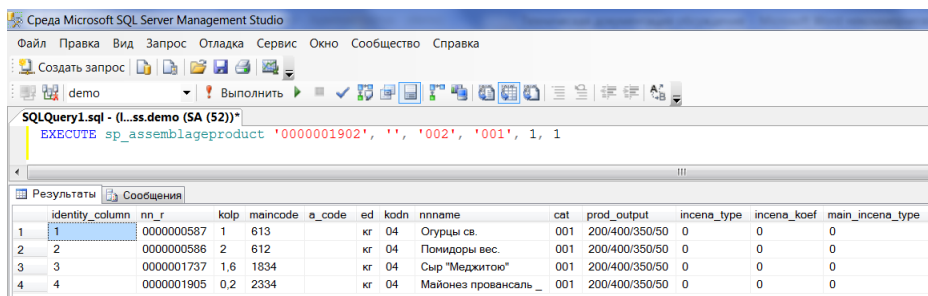
Процедура *sp\_assemblageproduct* расчет состава блюда на указанную дату на указанном складе.

Имя параметра	Тип параметра	Комментарий
@nn	char(10)	Код блюда из справочника ТМЦ, для которого производится расчет.
@date	datetime	Дата расчета. Если не указана, то используется текущая дата.
@sklad	char(3)	Код склада, для которого будет производиться расчет наличия ингредиентов.
@account_	char(3)	Код модели учета для расчета.
@f_make	int	1 – если требуется калькуляция “сборка”, иначе – 0
@mainkolp	float	Количество блюда, для которого необходимо произвести расчет.
@f_virtualcalc	int	Признак виртуальной калькуляции, которая состоит из одного передаваемого продукта.
@f_notCreateSaldo	int	Признак того, что не надо создавать собственную временную таблицу для поддержки остатков во время расчета.

**Примечание.** Данная процедура полностью обрабатывает калькуляции всех уровней ингредиентов, осуществляет все проверки наличия на складе, производит автоматически альтернативные замены. Следует обратить внимание, что результат всегда возвращается на одну единицу блюда. Параметр *@mainkolp* служит только для расчета ингредиентов, которые могут быть автоматически собраны при недостатке наличия на складе.

Выполним следующий запрос. Он вернет состав блюда на основе действующей калькуляции.

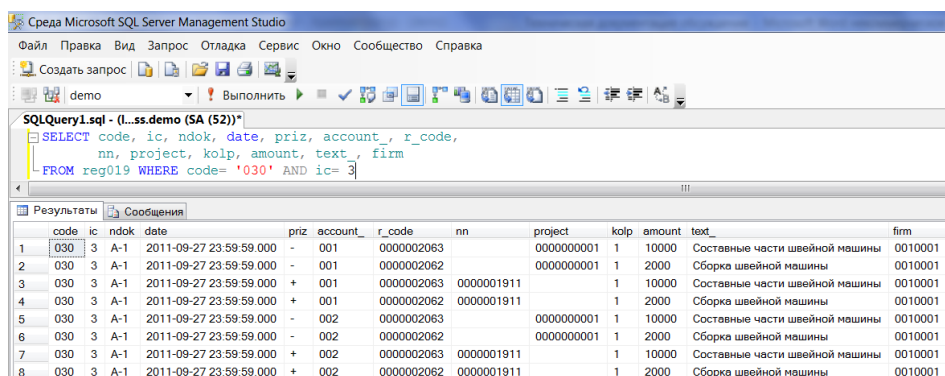
```
EXECUTE sp_assemblageproduct '0000001902', '', '002', '001', 1, 1
```



**Примечание.** Поле *identity\_column* в результате запроса служит для сортировки состава в порядке следования в калькуляции.

Рассмотрим содержимое этого регистра, но как регистра учета состава объектов ОС. Для этого выполним следующий запрос.

```
SELECT code, ic, ndok, date, priz, account_, r_code, nn, project, kolp,
       amount, text_, firm
FROM reg019 WHERE code= '030' AND ic= 3
```



Рассмотрим первую строку результата запроса.

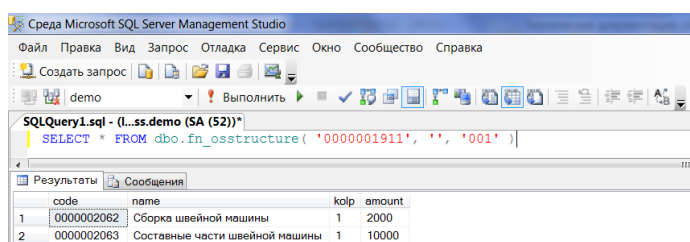
Имя поля	Значение	Комментарий
code	030	Код типа документа Формирование стоимости ОС.
ic	3	Идентификатор документа. Равно полю identity_column в таблице spr030.
ndok	A-1	Номер документа
date	2011-09-27 23:59:59.000	Дата документа.
priz	+	Для документов, которые увеличивают стоимость объекта значение этого поля равно '+'.
account_	001	Код модели учета, по которой создана запись.
r_code	0000002063	Код составной части из справочника ТМЦ или Услуг.
nn		При списании не указывается объект ОС, на который списаны составные части.
project	0000000001	Проект, с которого были списаны составные части.
kolp	1	Количество составных частей данного вида.
amount	10000	Сумма указанного количества составных частей.
text_	Составные части швейной машины	Наименование составной части объекта ОС.
firm	0010001	Код организации из документа.

*Функция fn\_osstructure вычисление состава объекта ОС на указанную дату.*

Имя параметра	Тип параметра	Комментарий
@os	char(10)	Код объекта из справочника ОС, для которого необходим состав.
@date	datetime	Дата расчета.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос

```
SELECT * FROM dbo.fn_osstructure( '0000001911', '', '001' )
```



Функция *fn\_projectstructure* вычисление состава проекта на указанную дату.

Имя параметра	Тип параметра	Комментарий
@project	char(10)	Код проекта из справочника проектов, для которого необходим состав.
@date	datetime	Дата расчета. Если не указана, то считается равной текущей дате.
@account_	char(3)	Код модели учета для расчета.

Выполним следующий запрос. Он вернет состав проекта, который был использован в приходной накладной № А-0.

```
SELECT * FROM dbo.fn_projectstructure( '0000000002', '', '001' )
```

