**Запросы.** **Терминология и обозначения**

- "Запрос" - объект метаданных типа "Запрос";

- Текст запроса - текстовое описание запроса к ИБ на специализированном языке запросов;

- Выборка - набор данных, полученных в результате выполнения запроса;

- Группировка данных - накопление различных значений одного и того же типа данных.

- Отчёт - совокупность текста запроса, процедуры его выполнения и интерпретации выбранных данных.

Введение

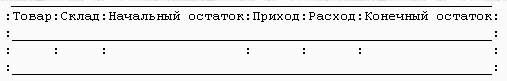
С чего начать

Начнем с начала

Для начала, прежде чем проектировать запрос, необходимо задаться вопросом: "Что же я хочу в результате получить?". Да, да, именно так все и начинается, с тривиальной "постановки задачи".

**Постановка задачи**

Предположим, я хочу получить "Данные об оборотах и остатках товаров на складах за какой-то период времени" в виде вот такой таблицы:



Формирование текста запроса

Уже изначально, в постановке задачи прозвучали все необходимые данные для построения запроса, такие как: обороты, остатки, товары, склады и период времени . Начнем по порядку:

1. "... за какой-то период времени". Значит, мы можем описать период времени, за который запрос будет извлекать информацию. Отразим это:

Период с '98' по '98'

Правды ради, следует отметить, что задание конкретных дат интервала запроса не слишком удобно в большинстве случаев, поэтому создадим в модуле отчета две внешние переменные, определяющие дату начала и конца запроса - ДатаС и ДатаПо :

Период с ДатаС по ДатаПо;

2. "Данные об оборотах и остатках товаров ...". Значит, мы хотим видеть перечень товаров. Ну что ж, давайте так и сделаем - объявим переменную запроса - Товар

Товар =

3. "Данные об оборотах и остатках товаров на складах ...". Для каждого товара хотим отобразить склад, на котором он хранится. Для этого объявляем переменную Склад :

Склад =

4. "Данные об оборотах ". Во-первых, необходимо определиться, какие конкретные обороты мы хотим отразить: денежные, количественные. Давайте определим

Количество =

Поскольку нам нужны данные об оборотах, то следует объявить и сами функции накопления:

Приход = Функция Приход(Количество);

Расход = Функция Расход(Количество);

Данные об оборотах можно получить различными способами – из документов, обходя все или определенные виды документов за конкретный период времени, и из регистров.

5. "Данные об остатках ". Объявим функции остатков:

НачОст = Функция НачОст(Количество);

КонОст = Функция КонОст(Количество);

Данные об остатках из документов уже не получишь, для этой цели предназначены регистры остатков. Следовательно, в работе запроса мы будем использовать регистры. Этот вывод существенно помогает в проектировании запроса, так как мы теперь в состоянии корректно описать переменные, зная, что учет оборотов и остатков ведется по регистру "Товарные запасы":

Товар = Регистр.ТовЗап.Товар;

Склад = Регистр.ТовЗап.Склад;

Количество = Регистр.ОстаткиТоваров.ОстатокТовара;

6. Если мы хотим получить перечень товаров, то нам необходимо эти товары группировать, то есть отображать различные виды товаров. Для чего мы заводим группировку по переменной Товар :

Группировка Товар;

7. Те же самые резоны приводят нас к созданию группировки по переменной Склад :

Группировка Склад;

8. Теперь соберем полный текст запроса, который можно присвоить какой-нибудь переменной модуля отчёта, скажем, ТекстЗапроса:

ТекстЗапроса =

"Период с ДатаС по ДатаПо

|Товар = Регистр.ТовЗап.Товар;

|Количество = Регистр.ТовЗап.ОстатокТовара;

|Приход = Функция Приход(Количество);

|Расход = Функция Расход(Количество);

|НачОст = Функция НачОст(Количество);

|КонОст = Функция КонОст(Количество);

|Группировка Товар;

|Группировка Склад;

|"//}}ЗАПРОС

;

Получившийся текст запроса - это только часть необходимой работы для получения отчёта, но это как раз та часть, которой мы посвятили эту главу методических рекомендаций.

Что происходит внутри

Как текст запроса превращается в выборку?

После того, как создан агрегатный объект типа "Запрос":

Запрос = СоздатьОбъект("Запрос");

сформулирован текст запроса и присвоен переменной ТекстЗапроса , запрос запускается на выполнение:

Запрос.Выполнить(ТекстЗапроса);

Что же происходит внутри этого метода? Как же там всё устроено?

Попадая в метод Выполнить , текст запроса превращается в выборку путём следующих метаморфоз:

- проверка синтаксиса;

- создание таблицы выборки;

- накопление данных.

Проверка синтаксиса

Правильность написания текста

Сначала проверяется правильность написания самого текста запросов - синтаксическая проверка запроса. Может быть пропущен символ ";", может быть неверно указан путь переменной запроса, в любом случае ошибки должны быть обнаружены и исправлены.

Для выявления синтаксических ошибок на стадии разработки (ещё при работе в конфигураторе) воспользуйтесь пунктом меню " Действия – Синтаксический контроль запросов... " , выберите запросы для проверки и нажмите кнопку OK.

Поскольку речь зашла о синтаксисе, обсудим его мало освещенные аспекты.

Объявление внутренних переменных запроса

достаточно подробно описано в книге "Описание встроенного языка. Часть 2". Но хотелось бы ещё раз обратить внимание на то, что описание переменной обуславливает характер обхода ИБ для формирования выборки. Мы можем описать переменную Товар различными способами:

Товар = Справочник.Товар.ТекущийЭлемент;

или

Товар = Документ.РасходнаяНал.Товар;

или

Товар = Регистр.ОстаткиТоваров.Товар;

и во всех трех случаях запрос будет вести себя по-разному. В первом случае запрос пройдет по всем элементам справочника Товар и выдаст полный его перечень. Во втором случае запрос пройдет по документам типа РасходнаяНал и выберет только те товары, которые были проданы по этим документам. А в третьем случае запрос пройдет по движениям или по остаткам регистра (в зависимости от того, какие функции были применены в запросе) и соберет те товары, которые участвовали в этих движениях или присутствовали в остатках.

Если переменную описывают путем с более чем одной ссылкой на объект типа справочник, документ, план счетов и т.д., то говорят, что переменная описана ДЛИННЫМ ПУТЕМ, например

Валюта = Регистр.ОстаткиТоваров.Товар.ВалютаРаботы;

Адрес = Документ.РасходнаяНал.Клиент.Адрес;

Так же хотелось обратить внимание на следующие особенности синтаксиса. При работе с регистрами имеется возможность обращения не только к документу, вызвавшему движение регистра:

Док = Регистр.ИмяРегистра.ТекущийДокумент;

но и к общим реквизитам документов:

Реквизит = Регистр.ИмяРегистра.ТекущийДокумент. ИмяРеквизита;

и к самим полям документов. Но для этого необходимо указать, из каких типов документов необходимо выбирать значения:

Поле = Регистр.ИмяРегистра.ТекущийДокумент. ИмяДокумента.ИмяПоля;

При описании поля документа нельзя не указывать имени самого документа, даже если такое поле встречается в каждом документе конфигурации. Для решения этой проблемы нужно либо объявить поле общим реквизитом документов, либо перечислить в описании переменной все необходимые типы документов:

Поле = Регистр.ИмяРегистра.ТекущийДокумент.

ИмяДокумента1.ИмяПоля,

Регистр.ИмяРегистра.ТекущийДокумент. ИмяДокументаN.ИмяПоля;

Для ИБ в формате .DBF не следует злоупотреблять длинными путями, так как это увеличивает время формирования выборки, ведь запросу приходится считывать не только реквизит указанного документа или регистра, но и по ссылке на объект считывать сам этот объект для извлечения данных. Что нельзя сказать о ИБ в формате SQL. Объект Запрос формирует запрос к SQL серверу, и чем больше работы будет переложено на плечи SQL сервера, тем быстрее будет формироваться выборка.

Оператор Период С

задает временной интервал выборки данных. Параметрами этого предложения могут быть:

- Дата - в формате встроенного языка '64', '1995'.

Внешняя переменная - глобальная переменная модуля отчёта или идентификатор элемента управления формы отчёта типа Дата или Документ . Если переменная - типа Документ , то интервал запроса устанавливается на дату и время указанного документа.

- Выражение встроенного языка. Задается в круглых или фигурных скобках и может иметь тип, как Дата, так и Документ, например:

Период с (ДатаС + 1) по (ПолучитьДокументТА());.

Оператор Условие

Задаёт условие включения данных в выборку. Для оптимизации времени выполнения запроса следует знать, что в 1С:Предприятии версии 7.7 запрос сам выполняет, а не вызывает для этого исполнительную среду встроенного языка, условия вида:

ЛеваяЧастьУсловия знакСравнения ПраваяЧастьУсловия

где,

ЛеваяЧастьУсловия и ПраваяЧастьУсловия могут быть константой, переменной запроса или внешней переменной;

знакСравнения - логический оператор (">", "<", "=", "<>", ">=", "<=", "в", "in").

Ниже приведен пример запроса с элементарными условиями:

"//{{Запрос(ЭлементарноеУсловие)

|Период с ДатаС по ДатаПо;

|Товар = Документ.ПриходнаяНал.Товар;

|Дата = Документ.ПриходнаяНал.ДатаДок;

|КолВо = Документ.ПриходнаяНал.Количество;

|Группировка Товар упорядочить по Товар.Наименование;

|Условие(Дата <> ИсключитьДату);

|Условие(Товар = ВыбранныйТовар);

|Условие(КолВо >= 43);

|Условие((Дата <> ИсключитьДату) ИЛИ (КолВо >= 43))

"//}}

;

Следует обратить внимание на то, что отдельные условия в

запросе объединяются по "И", то есть запросы

Условие(ВыходнойДень(Дата) = 0);

Условие(Товар в ВыбранныйТовар);

и

Условие((ВыходнойДень(Дата) = 0) И (Товар в ВыбранныйТовар)

= 1));

будут выполняться одинаково.

Если условие не подходит под определение элементарного, то оно выполняется механизмом внутреннего языка системы 1С:Предприятие.

Если существует возможность разбить сложное условие на элементарные, то так и следует поступить. Допустим, что нам необходимо применить условие вида:

"//{{Запрос(ПреобразованиеУсловия\_Было)

|//...

|Условие(Флаг = Перечисление.Булево.Да);

|//...

"//}}

;

, то для оптимизации времени исполнения запроса следует, объявить глобальную переменную:

Перем True;

а в конце модуля определить её как

True = Перечисление.Булево.Да;

и применить эту переменную для формирования элементарного условия:

"//{{Запрос(ПреобразованиеУсловия\_Стало)

|//...

|Условие(Флаг = True);

|//...

"//}}

;

Следует обратить особое внимание на то, что оператор принадлежности (в/in), не поддерживается внутренним языком 1С:Предприятия, поэтому оператор принадлежности должен использоваться только в элементарных условиях вычисляемых самим Запросом. И поэтому следующий пример будет выдавать ошибки при выполнении запроса:

"//{{ЗАПРОС(ОшибкаОператораПринадлежности)

|//...

|Условие(Товар.Код в ВыбТовар.Код);

|//...

"//}}

;

Оператор Группировка

в 1С:Предприятии версии 7.5 обогатился следующими реквизитами Все и Все ВошедшиеВЗапрос . Эти реквизиты созданы для того, чтобы давать возможность создавать табличные отчёты, разворачиваемые по горизонтали и вертикали. Для иллюстрации вышесказанного, обратимся к классике - отчёту "Остатки товаров на складах".

Далее приведен текст процедуры программного модуля, реализующего формирование отчёта:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Процедура формирования отчета//

Процедура ПоСкладам()

Перем Запрос, ТекстЗапроса, Таблица;

//Создание объекта типа Запрос

Запрос = СоздатьОбъект("Запрос");

ТекстЗапроса =

"//{{ЗАПРОС(ПоСкладам)

|Скл = Регистр.ОстаткиТоваров.Склад;

|Товар = Регистр.ОстаткиТоваров.Товар;

|КолВо = Регистр.ОстаткиТоваров.ОстатокТовара;

|Группировка Товар Упорядочить По Товар.Код;

|Группировка Скл Упорядочить По Скл.Код Все;

|Функция КО = КонОст(КолВо);

|"//}}ЗАПРОС

;

// Если ошибка в запросе, то выход из процедуры

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

Возврат;

КонецЕсли;

// Подготовка к заполнению отчета данными запроса

Таблица=СоздатьОбъект("Таблица");

Таблица.ИсходнаяТаблица("ПоСкладам");

Таблица.ВывестиСекцию("Заголовок");

// Формирование шапки отчета

Таблица.ВывестиСекцию("Шапка|Боковик");

Счётчик = 0;

Пока Запрос.Группировка("Товар") = 1 Цикл

Пока Запрос.Группировка("Скл") = 1 Цикл

Таблица.ПрисоединитьСекцию("Шапка|Склад");

Счётчик = Счётчик + 1;

КонецЦикла;

Если Счётчик>0 Тогда

прервать;

КонецЕсли;

КонецЦикла;

Запрос.ВНачалоВыборки();

// Процесс формирования строки отчета

Пока Запрос.Группировка("Товар") = 1 Цикл

Если Запрос.ЭтоГруппа(1) = 1 Тогда

Таблица.ВывестиСекцию("Группа|Боковик");

Для i = 1 по Счётчик Цикл

КоличествоДляТаблицы = 0;

Таблица.ПрисоединитьСекцию("Группа|Склад");

КонецЦикла;

Иначе

Таблица.ВывестиСекцию("Товар|Боковик");

Пока Запрос.Группировка("Скл") = 1 Цикл

КоличествоДляТаблицы = Запрос.КО;

Таблица.ПрисоединитьСекцию("Товар|Склад");

КонецЦикла;

КонецЕсли;

КонецЦикла;

// Вывод заполненной формы отчета

Таблица.ТолькоПросмотр(1);

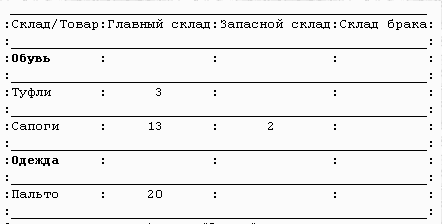
Таблица.Опции(0, 0, 5, 0);

Таблица.Показать("Остатки товаров на складах", "");

КонецПроцедуры

Выходная таблица отчёта имеет следующий вид:

Остатки товаров на складах



Следует упомянуть тот факт, что "Запрос" дает возможность доступа к переменным, группировкам и функциям по их именам. Значения упорядочивания хоть и находятся в выборке, но у их нет имени и следовательно, доступ к их значениям по имени невозможен. Для доступа к их значениям применяется метод ЗначениеУпорядочивания. Использование этого метода, может существенно ускорить время обхода выборки, по сравнению с использованием конструкций типа:

Запрос.ИмяГруппировки.ИмяПоля

из-за того, что значение ИмяПоля, система будет повторно выбирать из ИБ.

В 1С:Предприятии версии 7.7 появились новоые возможности по упорядочиванию группировок. Первая, это возможность упорядочивать значение группировки по функции. вторая, возможность новый реквизит предложения Группировка – без упорядочивания . Реквизит без упорядочивания удобен для пакетного режима выполнения запроса, когда не важно расположение элементов группировки, а необходимы только результаты. В этом случае сбор данных упорядочивания отключается полностью, в том числе и по умолчанию, что положительно сказывается на производительности.

Оператор Функция

В этом разделе хочется обратить внимание на то, что значения функций, объявленных в запросе, определены, накапливаются и сворачиваются по всем уровням группировок, начиная с самой старшей группировки - "Итого" и кончая самой младшей. Как иллюстрацию, приведем следующий пример:

Процедура ОстаткиТоваров()

Перем Запрос, ТекстЗапроса, Таб;

//Создание объекта типа Запрос

Запрос = СоздатьОбъект("Запрос");

ТекстЗапроса =

"//{{ЗАПРОС(ОстаткиТоваров)

|Период с ДатаС по ДатаПо;

|Товар = Регистр.ОстаткиТоваров.Товар;

|Склад = Регистр.ОстаткиТоваров.Склад;

|КолВо = Регистр.ОстаткиТоваров.ОстатокТовара;

|Группировка Товар упорядочить по

| Товар.Наименование;

|Группировка Склад упорядочить по

| Склад.Наименование;

|Функция Количество = КонОст(КолВо);

|"//}}ЗАПРОС

;

// Если ошибка в запросе, то выход из процедуры

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

Возврат;

КонецЕсли;

// Подготовка к заполнению выходных форм данными

Таб = СоздатьОбъект("Таблица");

Таб.ИсходнаяТаблица("ОстаткиТоваров");

// Заполнение полей "Заголовок"

Таб.ВывестиСекцию("Заголовок");

Состояние("Заполнение выходной таблицы...");

Пока Запрос.Группировка("Товар") = 1 Цикл

// Заполнение полей Товар

Таб.ВывестиСекцию("Товар");

Пока Запрос.Группировка("Склад") = 1 Цикл

// Заполнение полей Склад

Таб.ВывестиСекцию("Склад");

КонецЦикла;

КонецЦикла;

// Заполнение полей "Итого"

Таб.ВывестиСекцию("Итого");

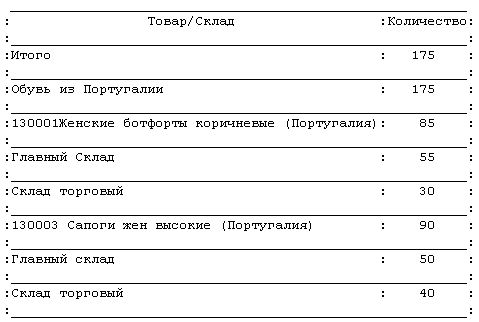
// Вывод заполненной формы

Таб.Опции(1, 0, 1, 0);

Таб.Показать("ОстаткиТоваров", "");

КонецПроцедуры

В результирующей таблице мы видим, что значение функции определено для Результирующей записи - "Итого", для группы справочника Товары, для элемента справочника Товары, для элемента справочника Склад. То есть, значение функции определено на всех уровнях вложенности.



Создание таблицы выборки

Формат хранения временных файлов

Запрос создает временные файлы для накопления и хранения выборки на компьютере, с которого этот запрос запустили. Формат хранения временных файлов .DBF - .CDX.

Структура записи таблицы выборки

Структура записи таблицы выборки следующая:

- Поля группировок . Поля группировок идут последовательно в порядке их объявления в тексте запроса, и этот порядок крайне важен. Поле группировки состоит из полей упорядочивания и поля значения. По полям упорядочивания и полям значения строятся индексы, и следует иметь в виду, что максимальная длина ключа .CDX .файла равна 240 байтам. Если вы заводите большое количество группировок или большое число полей упорядочивания – достичь этой величины не составляет труда. При превышении максимально допустимого значения механизм запросов пропорционально уменьшает длину строковых полей упорядочивания, входящих в ключ. Если же длина ключа превышает 240 байт, а возможность сократить её за счёт строковых полей отсутствует, то обработка запроса прекратится с ошибкой "Длина индекса превышает максимальную длину и не может быть уменьшена".

- Поля функций накопления . Это самые обыкновенные числовые поля с максимальной длиной. Эти поля не входят в индексы.

- Поля внутренних переменных запроса , по которым не велись ни группировки, ни накопление. Такие переменные нужны для выборки данных из объектов группировки, по которым не нужно вести ни группировку, ни упорядочивание, а значение необходимо.

Накопление данных

После того, как мы узнали, что представляет собой запись таблицы выборки, мы наконец-то можем поговорить о том, как накапливаются данные, подчёркиваю, именно, накапливаются . Как говорилось выше, обход ИБ запросом зависит от определения путей переменных. Если переменная Товар в тексте запроса определена следующим образом:

Период с ДатаС по ДатаПо;

Товар = Документ.РасходнаяНал.Товар,

Документ.РасходнаяКредит.Товар;

Склад = Документ.РасходнаяНал.Склад,

Документ.РасходнаяКредит. Склад;

КолВо = Документ.РасходнаяНал.Количество,

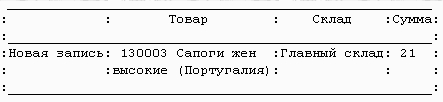
Документ.РасходнаяКредит. Количество

Группировка Товар;

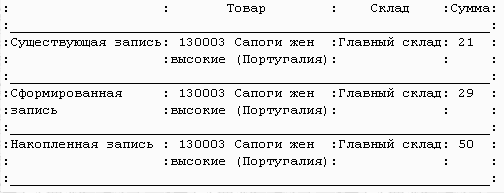
Группировка Склад;

Сумма = Функция Сумма(КолВо);

то запрос сначала обойдет все документы вида РасходнаяНал за указанный период, а затем - все документы вида РасходнаяКредит . И каждый раз запрос будет формировать запись, заполняя поля группировок, функций и внутренних переменных. После формирования записи, запрос будет добавлять её в таблицу выборки. Если полученного сочетания значений группировок нет, запрос добавляет запись в таблицу выборки:



а если такое сочетание значений группировок существует, то запись накапливается :



Если в записи не заполнено какое-либо поле группировок, то запись не считается сформированной, не записывается и не участвует в накоплении значений.

Использование граф отбора

Использование граф отбора может существенно убыстрить время формирования запроса, но для того, чтобы воспользоваться графами отбора, запрос должен удовлетворять следующим требованиям:

- в запросе должна быть объявлена переменная, пути которой включены в графу отбора;

- должно быть задано элементарное условие;

- условие должно быть со знаком сравнение "=".

Товар = Регистр.ИмяРегистра.Поле;

Условие(Товар = ВыбТовар); // (+)

Условие(Товар <> ВыбТовар); // (-) не знак =

Условие((Товар) = ВыбТовар); // (-) не элементарное

// условие

где ВыбТовар - внешняя переменная запроса.

У объекта метаданных типа "Запрос" существует метод ИспользоватьГрафуОтбора , который может управлять механизмом выборки данных с использованием графы отбора.

Параметр метода ИспользоватьГрафуОтбора может принимать следующие значения:

"\*" - автоматический выбор графы отбора. Если в тексте запроса описаны элементарные условия со знаком "равно", то запрос просматривает переменную, фигурирующую в условии, и пытается найти для неё соответствующую графу отбора, и, если такая переменная найдена, то включается механизм выборки данных с использованием графы отбора. Когда в тексте запроса объявлены несколько условий, удовлетворяющих критериям включения механизма выборки данных с использованием графы отбора, перед запросом встает задача выбора одной единственной графы отбора из списка возможных. Запрос в своих оценках выбора графы отбора ориентируется на количество элементов справочников и перечислений. Для других типов оценка не производится, и, при желании, можно задать графу отбора вручную.

имя графы отбора . Графа отбора задаётся вручную. Если она задана неверно, то тогда при выполнении запроса выведется сообщение: "Нет переменной, удовлетворяющей заданной графе отбора".

" " - отказ от использования графы отбора.

Особенности использования запросов

Метод Выполнить может быть применен к объекту "Запрос" неограниченное количество раз. Приведем тому пример:

Процедура Сформировать()

Перем Запрос, ТекстЗапроса, Таб;

Запрос = СозадтьОбъект("Запрос");

ТекстЗапроса =

"//{{ЗАПРОС(ТекстЗапроса1)

|...

"//}}ЗАПРОС

;

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

Возврат;

КонецЕсли;

// Цикл обработки запроса

...

"//{{ЗАПРОС(ТекстЗапроса2)

|...

"//}}ЗАПРОС

;

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

Возврат;

КонецЕсли;

// Цикл обработки запроса

КонецПроцедуры

При повторном вызове метода Выполнить , объект Запрос инициализирует внутренние структуры, и доступ к переменным, объявленным в предыдущем запросе и выбранным данным становится, не возможен.

Особенности позиционирования >. После формирования выборки, запрос позиционируется на первую запись ("Итого") – в начало выборки, с которой начинается обход выбранных значений. Вернуться в начало выборки, можно двумя способами. В цикле обхода пройти всю выборку до конца;

// Формирование шапки отчета

Таблица.ВывестиСекцию("Шапка|Боковик");

Счётчик = 0;

Пока Запрос.Группировка("Товар") = 1 Цикл

Если Счётчик = 0 Тогда

Пока Запрос.Группировка("Скл") = 1 Цикл

Таблица.ПрисоединитьСекцию("Шапка|Склад");

Счётчик = Счётчик + 1;

КонецЦикла;

КонецЕсли;

КонецЦикла;

// Процесс формирования строки отчета

Пока Запрос.Группировка("Товар") = 1 Цикл

Если Запрос.ЭтоГруппа(1) = 1 Тогда

Таблица.ВывестиСекцию("Группа|Боковик");

Для i = 1 по Счётчик Цикл

КоличествоДляТаблицы = 0;

Таблица.ПрисоединитьСекцию("Группа|Склад");

КонецЦикла;

Иначе

Таблица.ВывестиСекцию("Товар|Боковик");

Пока Запрос.Группировка("Скл") = 1 Цикл

КоличествоДляТаблицы = Запрос.КО;

Таблица.ПрисоединитьСекцию("Товар|Склад");

КонецЦикла;

КонецЕсли;

КонецЦикла;

или воспользоваться методом "Запроса" - ВначалоВыборки:

Таблица.ВывестиСекцию("Шапка|Боковик");

Счётчик = 0;

Пока Запрос.Группировка("Товар") = 1 Цикл

Пока Запрос.Группировка("Скл") = 1 Цикл

Таблица.ПрисоединитьСекцию("Шапка|Склад");

Счётчик = Счётчик + 1;

КонецЦикла;

Если Счётчик > 0 Тогда

прервать;

КонецЕсли;

КонецЦикла;

Запрос.ВНачалоВыборки();

// Процесс формирования строки отчета

Пока Запрос.Группировка("Товар") = 1 Цикл

Если Запрос.ЭтоГруппа(1) = 1 Тогда

Таблица.ВывестиСекцию("Группа|Боковик");

Для i = 1 по Счётчик Цикл

КоличествоДляТаблицы = 0;

Таблица.ПрисоединитьСекцию("Группа|Склад");

КонецЦикла;

Иначе

Таблица.ВывестиСекцию("Товар|Боковик");

Пока Запрос.Группировка("Скл") = 1 Цикл

КоличествоДляТаблицы = Запрос.КО;

Таблица.ПрисоединитьСекцию("Товар|Склад");

КонецЦикла;

КонецЕсли;

КонецЦикла;

Метод ВначалоВыборки можно вызывать на любом уровне вложенности циклов обработки выборки, для перехода в ее начало. Количество обходов выборки ничем не ограничено.

**Методы оптимизации запросов**

Общие положения написания запросов

Примером неправильной работы с данными в 1С:Предприятиии, есть и остается избыточное обращение к базе данных, вне зависимости от того, в каком формате она хранится.

Как избежать лишних обращений к ИБ, и тем самым существенно уменьшить время формирования отчета проиллюстрируем на примере "Оптимизация Отчета".

Закладка 100%

ТекстЗапроса =

"//{{ЗАПРОС(Сформировать100)

|Период с ВыбНачПериода по ВыбКонПериода;

|Товар = Документ.РасхНакл.Товар;

|Количество = Документ.РасхНакл.Количество;

|Функция КоличествоСумма = Сумма(Количество);

|Группировка Товар;

|Условие(Товар в ВыбТовар);

|Условие(Товар.ВидТовара = ВыбВидТовара);

|"//}}ЗАПРОС

;

При написании текста запроса необходимо учитывать, что Запрос при формировании выборки сам отрабатывает только элементарные условия , не запуская при этом исполнительную среду встроенного языка, что значительно экономит время. Если есть возможность привести условие к элементарному - это необходимо сделать.

"//{{ЗАПРОС(Сформировать15)

|Период с ВыбНачПериода по ВыбКонПериода;

|Товар = Документ.РасхНакл.Товар;

|ВидТовара = Документ.РасхНакл.Товар.ВидТовара;

|Количество = Документ.РасхНакл.Количество;

|Функция КоличествоСумма = Сумма(Количество);

|Группировка Товар;

|Условие(Товар в ВыбТовар);

|Условие(ВидТовара=ВыбВидТовара);

|"//}}ЗАПРОС

;

Так как выборка формируется на компьютере пользователя, то все данные, которые необходимы пользователю для построения отчета, должны быть, по возможности, получены посредством запроса. В таблице выводится значение кода товара – для этого в запросе необходимо ввести переменную КодТовара . Тогда для формирования отчета, отпадет необходимость, каждый раз обращаться к ИБ. Следует не забывать про методы Запроса - ЗначениеУпорядочивания и ЭтоГруппа , которые опять же извлекают данные из выборки, минуя ИБ, что заметно уменьшает время формирования запроса.

"//{{ЗАПРОС(Сформировать3)

|Период с ВыбНачПериода по ВыбКонПериода;

|Товар = Документ.РасхНакл.Товар;

|ВидТовара = Документ.РасхНакл.Товар.ВидТовара;

|КодТовара = Документ.РасхНакл.Товар.Код;

|Количество = Документ.РасхНакл.Количество;

|Функция КоличествоСумма = Сумма(Количество);

|Группировка Товар;

|Условие(Товар в ВыбТовар);

|Условие(ВидТовара=ВыбВидТовара);

|"//}}ЗАПРОС

;

В отчете "Оптимизация Отчета" время выполнение самой медленной версии взято за 100%, остальные версии берутся как процент от нее. Следование правилам, дало возможность уменьшить время выполнения запроса почти на 2 порядка!

Особенности использования запросов для получения информации из справочников

В случае применения объекта "Запрос" для выборки информации из справочников 1С:Предприятия необходимо учитывать следующую особенность. При обработке справочника объект "Запрос" не обрабатывает группы справочника. То есть он не использует их в качестве исходных данных для получения первичной выборки. При обработке уже полученных записей запрос добавляет группы для выбранных записей - элементов, если существует группировка по переменной запроса типа "Справочник". Такой способ обработки является стандартным, и Запрос выполняет ее так же, как он это делает, например, для реквизитов документа имеющих тип "Справочник". То есть, если в запросе по документам использовать группировку по реквизиту "Товар" документа "Счет", то в полученном отчете можно получить записи и по группам товаров, вошедших в запрос. Аналогично, если в запросе по справочнику получать в качестве группировки текущий элемент, то в полученный отчет будут включены записи, соответствующие группам отобранных запросом элементов. Однако так как сами группы не обрабатываются запросом при заполнении таблицы выборки, то в отчет не попадут те группы, которые не имеют элементов, или группы, элементы которых не попали в выборку. Соответственно объект "Запрос" не может применяться в тех случаях, когда нужно обрабатывать собственно группы, или получать все элементы, включая группы. С другой стороны, так как в большинстве случаев должны выбираться непосредственно элементы, запрос может быть успешно применен для обработки справочника. В том числе, Запрос позволяет существенно ускорить выборку элементов по условию, при работе с базой данных в формате SQL.

При обработке с помощью запроса справочников, имеющих периодические реквизиты, следует учитывать, что значения периодических реквизитов выбираются только на конечную границу периода запроса. То есть, с помощью запроса нет возможности получить историю периодических реквизитов, а можно получить только срез значений на определенный момент.