

???????

- [Одноуровневые запросы](#)
- [Вложенные запросы, соединение, объединение, пакетные запросы](#)
- [Конструктор запросов](#)
- [Способы обхода результата запроса](#)

- упорядочивание результатов,
- автоупорядочивание,
- описание итогов.

Для реальных полей поддерживается * для выбора всех полей. Виртуальные поля нужно задавать явно.

ВЫБРАТЬ

Справочник.Клиенты.*

Полный текст с получением столбцов в запросе:

```

&НаСервереБезКонтекста
Процедура ПолучитьВсюТаблицуНаСервере()
□Запрос = Новый Запрос;
□Запрос.Текст = "ВЫБРАТЬ
□□ | □Справочник.Филиалы.*";
□Результат = Запрос.Выполнить();
□Если не Результат.Пустой() Тогда
□□ВыборкаДетальныеЗаписи = Результат.Выбрать();
□□РезультатЗапроса = ВыборкаДетальныеЗаписи.Владелец();
□□ПолнаяСтрокаОвета = "";
□□Для Каждого Кол ИЗ РезультатЗапроса.Колонки Цикл
□□□ПолнаяСтрокаОвета = ПолнаяСтрокаОвета + Кол.Имя + " ";
□□КонецЦикла;
□□Сообщить(ПолнаяСтрокаОвета);
□□Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
□□□ПолнаяСтрокаОвета = "";
□□□Для Каждого Кол ИЗ РезультатЗапроса.Колонки Цикл
□□□□ПолнаяСтрокаОвета = ПолнаяСтрокаОвета + ВыборкаДетальныеЗаписи[Кол.Имя] + " ";
□□□КонецЦикла;
□□□Сообщить(ПолнаяСтрокаОвета);
□□КонецЦикла;
□КонецЕсли;
КонецПроцедуры

```

Настройка параметров запроса

```
Запрос.УстановитьПараметр("НачалоПериода", '20140101000000');
```

Условия и преобразования в ВЫБРАТЬ

Операция выбора (ВЫБОР (КОГДА ТОГДА) ИНАЧЕ КОНЕЦ). После ключевого слова КОГДА записывается условие выбора, после ключевого слова ТОГДА значение поля выборки в случае, если условие истинно. В общем случае в операции выбора может указываться неограниченное количество альтернативных одиночных выборов КОГДА ... ТОГДА. Значение выражения, указанного после слова ИНАЧЕ, используется в качестве результата операции выбора в том случае, если ни одно из ранее указанных условий выбора не было выполнено.

ВЫБРАТЬ

Товары.Наименование КАК Наименование,

ВЫБОР

КОГДА (Товары.Производитель) ЕСТЬ NULL ТОГДА "NULL"

ИНАЧЕ Товары.Производитель

КОНЕЦ КАК Производитель

ИЗ

Справочник.Товары КАК Товары

Также в ВЫБРАТЬ могут содержаться следующие элементы:

- Литералы типов: число, строка (в кавычках), булево (значения Истина и Ложь), Null, Неопределено. Чтобы указать литерал типа Дата, можно воспользоваться ключевым словом языка запросов ДАТАВРЕМЯ или передать дату через параметр запроса.
- Арифметические операции (+, -, /, *). Операция получения остатка % в языке запросов не поддерживается.
- Операцию конкатенации строк (+). Операцию конкатенации нельзя использовать для виртуальных полей.
- Встроенные функции языка запросов (ДЕНЬ, МЕСЯЦ, ГОД и т. д.).
- Агрегатные функции (СУММА, МИНИМУМ, МАКСИМУМ, СРЕДНЕЕ, КОЛИЧЕСТВО).
- Операцию выбора ВЫБОР – позволяет получить одно из возможных значений в соответствии с указанными условиями.
- Операцию приведения типов ВЫРАЗИТЬ – позволяет привести значение составного типа к одному из составляющих это значение типов. А также функцию ВЫРАЗИТЬ() используют для получения результатов нужной длины и точности.

ВЫБРАТЬ

Товары.Наименование + " (" + Товары.Код + ")" КАК Товар,

ВЫБОР

КОГДА Товары.ЭтоГруппа = ИСТИНА ТОГДА "Это группа"

ИНАЧЕ "Это элемент"

КОНЕЦ КАК ПризнакГруппы

ИЗ

ВЫБРАТЬ

НАЧАЛОПЕРИОДА (ОстаткиТоваров.Период, НЕДЕЛЯ) КАК Период,

ВЫБОР

КОГДА (ОстаткиТоваров.Регистратор ССЫЛКА Документ.ПриходнаяНакладная)

ТОГДА ВЫРАЗИТЬ (ОстаткиТоваров.Регистратор КАК

Документ.ПриходнаяНакладная).Поставщик

КОГДА (ОстаткиТоваров.Регистратор ССЫЛКА Документ.РасходнаяНакладная)

ТОГДА ВЫРАЗИТЬ (ОстаткиТоваров.Регистратор КАК

Документ.РасходнаяНакладная).Покупатель

КОНЕЦ КАК Контрагент

ИЗ

РегистрНакопления.ОстаткиТоваров КАК ОстаткиТоваров

ГДЕ

ОстаткиТоваров.Период МЕЖДУ &ДатаНачала И &ДатаОкончания

Упорядочивание результата.

Раздел УПОРЯДОЧИТЬ ПО, обязательно добавлять для определенности. Есть ВОЗР (по умолчанию, можно не указывать) и УБЫВ. Можно использовать поля вне блока ВЫБРАТЬ.

ВЫБРАТЬ

Цены.Товар КАК Товар,

Цены.Цена КАК Цена

ИЗ

РегистрСведений.Цены КАК Цены

УПОРЯДОЧИТЬ ПО

Цены.Период УБЫВ,

Цена

По типу Ссылка управляемо упорядочить нельзя. Однако с помощью автоупорядочивания можно вывести записи таблицы в наиболее естественном (ожидаемом пользователем) порядке. Для этого нужно упорядочить записи таблицы непосредственно по ссылочному полю, а затем использовать конструкцию АВТОУПОРЯДОЧИВАНИЕ

ВЫБРАТЬ

ЗаказТовара.Дата,

ЗаказТовара.Номер,

```
ЗаказТовара.Клиент,  
ЗаказТовара.СуммаЗаказа  
ИЗ  
Документ.ЗаказТовара КАК ЗаказТовара  
  
УПОРЯДОЧИТЬ ПО  
ЗаказТовара.Ссылка  
АВТОУПОРЯДОЧИВАНИЕ
```

Текстовое представление ссылочного поля

При выводе значения ссылочного поля для получения его представления выполняется дополнительный запрос к той таблице, на которую ссылается это ссылочное поле. В результате процесс замедляется. Для ускорения, нужно в запросе сразу получить текстовое представление ссылочного поля и затем уже его, а не саму ссылку, выводить в отчет или сообщение.

Каждая объектная таблица в информационной базе имеет виртуальное поле Представление. Это текстовое представление объекта. При получении данного поля запрос получает несколько полей, которые соответствуют прикладной сущности объекта, а при получении значения поля из результата запроса преобразовывает полученные значения в строку. Варианты ПРЕДСТАВЛЕНИЕ(Товар) аналогично Товар.Представление, но лучше первый.

```
ВЫБРАТЬ  
ОстаткиТоваров.Период,  
ПРЕДСТАВЛЕНИЕ(ОстаткиТоваров.Регистратор) КАК Регистратор,  
ОстаткиТоваров.Количество,  
ОстаткиТоваров.Сумма  
ИЗ  
РегистрНакопления.ОстаткиТоваров КАК ОстаткиТоваров
```

Но результат функции ПРЕДСТАВЛЕНИЕ() не может быть использован в выражении языка запросов, в условиях сравнения в предложении ГДЕ, в любых операциях.

Ограничение по количеству записей

```
ВЫБРАТЬ ПЕРВЫЕ 3
```

Непересекающаяся выборка

ВЫБРАТЬ РАЗЛИЧНЫЕ

ЗаказТовара.Клиент,

ЗаказТовара.Дата

ИЗ

Документ.ЗаказТовара КАК ЗаказТовара

Если в запросе указано ключевое слово РАЗЛИЧНЫЕ и в предложении УПОРЯДОЧИТЬ ПО указано поле, отсутствующее в списке выборки, то при выполнении такого запроса будет выдана ошибка.

Получение данных табличной части

ВЫБРАТЬ

СоставЗаказа.Товар,

СоставЗаказа.Количество,

СоставЗаказа.Сумма

ИЗ

Документ.ЗаказТовара.Состав КАК СоставЗаказа

Этот запрос выведет все табличные части всех документов Заказ товара. Для получения также информации о реквизитах документа используется ссылка. Можно также установить условие.

ВЫБРАТЬ

СоставЗаказа.Ссылка.Номер,

СоставЗаказа.Ссылка.Клиент,

СоставЗаказа.Товар,

СоставЗаказа.Количество,

СоставЗаказа.Сумма

ИЗ

Документ.ЗаказТовара.Состав КАК СоставЗаказа

ГДЕ

СоставЗаказа.Ссылка = &Документ

Можно получить табличную часть в виде вложенной таблицы и затем обойти ее в цикле:

&НаСервереБезКонтекста

Процедура ПолучитьТабличнуюЧастьНаСервере()

□Запрос = Новый Запрос;

□Запрос.Текст ="

□□| ВЫБРАТЬ

```

□□| Фирмы.Наименование КАК Название,
□□| Фирмы.КонтактныеЛица КАК Контакты
□□| ИЗ
□□| Справочник.Фирмы КАК Фирмы";
□Результат = Запрос.Выполнить();
□Если не Результат.Пустой() Тогда
□□ВыборкаДетальныеЗаписи = Результат.Выбрать();
□□Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
□□□ПолнаяСтрокаОвета = ВыборкаДетальныеЗаписи.Название;
□□□Сообщить(ПолнаяСтрокаОвета);
□□□ТекКонт = ВыборкаДетальныеЗаписи.Контакты.Выбрать();
□□□Пока ТекКонт.Следующий() Цикл
□□□□Сообщить(ТекКонт.Сотрудник)
□□□КонецЦикла;
□□КонецЦикла;
    КонецЕсли;
КонецПроцедуры

```

Запрос с одной и двумя вложенными таблицами:

```

// Первый вариант – одна вложенная таблица
ВЫБРАТЬ
    ЗаказТовара.Номер,
    ЗаказТовара.Клиент,
    ЗаказТовара.СуммаЗаказа,
    ЗаказТовара.Состав.(
        Товар,
        Количество,
        Сумма
    )
ИЗ
    Документ.ЗаказТовара КАК ЗаказТовара
// Второй вариант – две вложенные таблицы
ВЫБРАТЬ
    ЗаказТовара.Номер,
    ЗаказТовара.Клиент,
    ЗаказТовара.СуммаЗаказа,
    ЗаказТовара.Состав.Товар,
    ЗаказТовара.Состав.Сумма
ИЗ

```

Получение иерархической структуры

Ключевое слово ИЕРАРХИЯ после поля, по которому происходит сортировка, позволяет сортировать в соответствии с настроенными группами.

```
ВЫБРАТЬ
    Товары.Наименование КАК Наименование,
    Товары.ЭтоГруппа,
ИЗ
    Справочник.Товары КАК Товары
УПОРЯДОЧИТЬ ПО
    Наименование ИЕРАРХИЯ
```

Получение групп

```
// Условие без параметров
ВЫБРАТЬ
    Товары.*
ИЗ
    Справочник.Товары КАК Товары
ГДЕ
    Товары.ЭтоГруппа = ИСТИНА
```

Получение товаров определенной группы

```
ВЫБРАТЬ
    Товары.*
ИЗ
    Справочник.Товары КАК Товары
ГДЕ
    Товары.ЭтоГруппа = &ЭтоГруппа
    И Товары.Родитель = &Родитель
```

Получение корневых записей

```
ВЫБРАТЬ
    Товары.*
ИЗ
    Справочник.Товары КАК Товары
```

ГДЕ

Товары.Родитель = ЗНАЧЕНИЕ(Справочник.Товары.ПустаяСсылка

Получение записей принадлежащих группе

ВЫБРАТЬ

Товары.*

ИЗ

Справочник.Товары КАК Товары

ГДЕ

Товары.Ссылка В ИЕРАРХИИ (&ГруппаТоваров)

УПОРЯДОЧИТЬ ПО

Наименование ИЕРАРХИЯ

Ограничения на выборку и параметры запроса

ВЫБРАТЬ

Накладная.Дата КАК Дата,

Накладная.Номер КАК Номер,

Накладная.Поставщик

ИЗ

Документ.ПриходнаяНакладная КАК Накладная

ГДЕ

Дата >= ДАТАВРЕМЯ(2012, 11, 01)

ВЫБРАТЬ

Клиенты.Наименование КАК Наименование,

Клиенты.Адрес,

Клиенты.Телефон

ИЗ

Справочник.Клиенты КАК Клиенты

ГДЕ

Наименование ПОДОБНО "%Иван%"

В функции ПОДОБНО % - любое кол-во символов, _ - один символ,

Параметр запроса

```
Запрос = Новый Запрос;  
Запрос.Текст =  
"..."  
"ГДЕ"  
"И Задача.ГлавныйБП = &ГлавныйБП";  
Запрос.УстановитьПараметр("ГлавныйБП", СсылкаНаБизнесПроцесс);
```

Агрегатные функции и группировка

Агрегатные функции обычно вызываются вместе с группировкой. При отсутствии СГРУППИРОВАТЬ ПО все записи исходной таблицы будут сгруппированы в одну строку.

```
ВЫБРАТЬ  
    КОЛИЧЕСТВО(*) КАК Всего,  
    КОЛИЧЕСТВО(РАЗЛИЧНЫЕ ЗаказТовара.Клиент) КАК РазныеКлиенты  
ИЗ  
    Документ.ЗаказТовара КАК ЗаказТовара
```

```
ВЫБРАТЬ  
    НакладнаяСостав.Товар КАК Товар,  
    МИНИМУМ(НакладнаяСостав.Цена) КАК Минимум,  
    МАКСИМУМ(НакладнаяСостав.Цена) КАК Максимум,  
    СРЕДНЕЕ(НакладнаяСостав.Цена) КАК Среднее  
ИЗ  
    Документ.РасходнаяНакладная.Состав КАК НакладнаяСостав  
СГРУППИРОВАТЬ ПО  
    НакладнаяСостав.Товар
```

Всего 5 функций МИНИМУМ, МАКСИМУМ, СРЕДНЕЕ, КОЛИЧЕСТВО, СУММА. В предложении СГРУППИРОВАТЬ ПО должны указываться именно имена полей, а не их псевдонимы, определенные в запросе.

В случае группировки по значению нескольких полей нужно после ключевого слова СГРУППИРОВАТЬ ПО перечислить через запятую список полей группировки.

```
СГРУППИРОВАТЬ ПО  
    НакладнаяСостав.Ссылка.Покупатель,  
    НакладнаяСостав.Товар
```

ИМЕЮЩИЕ добавляет условие на значения агрегатных функций, применяемых к исходным записям с одинаковым значением поля группировки. Оно накладывает условие на записи, получившиеся в результате группировки.

ВЫБРАТЬ

НакладнаяСостав.Ссылка.Покупатель КАК Покупатель,
НакладнаяСостав.Товар КАК Товар,
СУММА(НакладнаяСостав.Количество) КАК Количество,
СУММА(НакладнаяСостав.Сумма) КАК Сумма

ИЗ

Документ.РасходнаяНакладная.Состав КАК НакладнаяСостав

СГРУППИРОВАТЬ ПО

НакладнаяСостав.Ссылка.Покупатель,
НакладнаяСостав.Товар

ИМЕЮЩИЕ

СУММА(НакладнаяСостав.Сумма) > 50000

Итоги

Итоги отличаются от группировки тем, что не сворачивают данные, а добавляют данные по итогам. Итоги можно считать не по всем полям.

ВЫБРАТЬ

НакладнаяСостав.Товар КАК Товар,
НакладнаяСостав.Количество КАК Количество,
НакладнаяСостав.Цена КАК Цена,
НакладнаяСостав.Сумма КАК Сумма

ИЗ

Документ.ПриходнаяНакладная.Состав КАК НакладнаяСостав

ИТОГИ

СУММА(Количество),
СРЕДНЕЕ(Цена),
СУММА(Сумма)

ПО

Товар

За счет ключевых слов ИЕРАРХИЯ и ТОЛЬКО ИЕРАРХИЯ можно считать итоги по иерархии (или только)

ВЫБРАТЬ

НакладнаяСостав.Товар КАК Товар,
НакладнаяСостав.Количество КАК Количество,
НакладнаяСостав.Цена КАК Цена,
НакладнаяСостав.Сумма КАК Сумма

ИЗ

Документ.ПриходнаяНакладная.Состав КАК НакладнаяСостав
ИТОГИ
СУММА(Количество) ,
СУММА(Сумма)
ПО
Товар ТОЛЬКО ИЕРАРХИЯ КАК ТоварТолькоИерархия

Если нужны итоги по нескольким полям, то их перечисляют через запятую в разделе ПО. Для расчета общих итогов используется слово ОБЩИЕ

ВЫБРАТЬ
НакладнаяСостав.Ссылка.Поставщик КАК Поставщик,
НакладнаяСостав.Товар КАК Товар,
НакладнаяСостав.Количество КАК Количество,
НакладнаяСостав.Сумма КАК Сумма
ИЗ
Документ.ПриходнаяНакладная.Состав КАК НакладнаяСостав
ИТОГИ
СУММА(Количество) ,
СУММА(Сумма)
ПО
ОБЩИЕ ,
Поставщик

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

Вложенные запросы

ВЫБРАТЬ

Расход.Ссылка КАК Документ,
Расход.Покупатель КАК Покупатель

ИЗ

Документ.РасходнаяНакладная КАК Расход

ГДЕ

Расход.Состав.Товар В

(

ВЫБРАТЬ

ПриходСостав.Товар КАК Товар

ИЗ

Документ.ПриходнаяНакладная.Состав КАК ПриходСостав

ГДЕ

ПриходСостав.Ссылка = &Документ

)

Вложенные запросы могут использоваться в аргументах конструктора виртуальных таблиц

ВЫБРАТЬ

Цены.Период,
Цены.Товар,
Цены.Цена

ИЗ

РегистрСведений.Цены.СрезПоследних(, Товар В

(

ВЫБРАТЬ

РасходСостав.Товар КАК Товар

ИЗ

Документ.РасходнаяНакладная.Состав КАК РасходСостав

ГДЕ

РасходСостав.Ссылка = &Документ

)

) КАК Цены

УПОРЯДОЧИТЬ ПО

Период

Соединения

Используются несколько источников данных, которые перечисляются после ключевого слова ИЗ. Обычно соединяются по ссылкам. Исходные таблицы запроса соединяются по условию связи. Условие связи источников запроса задается после ключевого слова ПО.

Внутреннее соединение. В результат запроса попадут записи из таблиц-источников, которые удовлетворяют заданному условию связи. Пересечение множеств, оператор И. Ключевое слово ВНУТРЕННЕЕ СОЕДИНЕНИЕ или СОЕДИНЕНИЕ.

ВЫБРАТЬ

Товары.Код,

Товары.Наименование,

Товары.Производитель,

Цены.Цена

ИЗ

Справочник.Товары КАК Товары

ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены

ПО Товары.Ссылка = Цены.Товар

Если будет найдено несколько записей, удовлетворяющих условию соединения, то в результат запроса будут включены все эти записи. Т е строчек будет несколько на одну запись.

Левое / правое внешнее соединение. При левом соединении попадут все строки левой таблицы, добавятся указанные столбцы из правой, если в правой есть данные - добавятся, иначе null. Правое аналогично.

ВЫБРАТЬ

Товары.Код,

Товары.Наименование,

Товары.Производитель,

Цены.Цена

ИЗ

Справочник.Товары КАК Товары

ЛЕВОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены
ПО Товары.Ссылка = Цены.Товар

Полное внешнее соединение. Добавляются указанные столбцы из обеих таблиц, попадают все строки из всех таблиц, если в какой-то нет данных - устанавливается NULL. Оператор ИЛИ.

При работе в клиент-серверном варианте с PostgreSQL производительность с конструкцией ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ снижается. Особенно когда в запросе встречаются две и более такие конструкции. Не рекомендуется использовать полное внешнее соединение в запросах.

Соединение нескольких таблиц:

```
ВЫБРАТЬ
    Товары.Наименование,
    ОстаткиТоваров.КоличествоОстаток,
    Продажи.КоличествоОборот
ИЗ
    Справочник.Товары КАК Товары
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиТоваров.Остатки КАК ОстаткиТоваров
    ПО Товары.Ссылка = ОстаткиТоваров.Товар
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК Продажи
    ПО Товары.Ссылка = Продажи.Товар
ГДЕ
    Товары.ЭтоГруппа = ЛОЖЬ
```

Здесь необходимо учитывать последовательность соединения и условие задачи, здесь начинает проявляться профессионализм составления оптимальных запросов.

При вложении запросов сначала выполняется самое глубокое вложение.

```
ВЫБРАТЬ
    Товары.Наименование,
    ОстаткиТоваров.КоличествоОстаток,
    Продажи.КоличествоОборот
ИЗ
    Справочник.Товары КАК Товары
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиТоваров.Остатки КАК ОстаткиТоваров
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК Продажи
    ПО ОстаткиТоваров.Товар = Продажи.Товар
    ПО Товары.Ссылка = ОстаткиТоваров.Товар
```

ГДЕ

Товары.ЭтоГруппа = ЛОЖЬ

Т е сначала левое соединение РегистрНакопления.ОстаткиТоваров.Остатки и РегистрНакопления.Продажи.Обороты, затем левое соединение Справочник.Товары и предыдущий результат.

Разыменованние. Если присутствует ссылочное поле на другой элемент, то в 1С вместо соединения рекомендуют использовать вложенное обращение к полям. Это эквивалентно левому соединению, но проще выглядит.

ВЫБРАТЬ

Событие.Дата КАК Дата,

Событие.ВидСобытия,

Событие.Клиент,

Событие.Клиент.Адрес КАК Адрес,

Событие.Клиент.Телефон КАК Телефон

ИЗ

Документ.Событие КАК Событие

УПОРЯДОЧИТЬ ПО

Дата

Объединения запросов.

При объединении каждый запрос получает данные независимо, то есть у каждого из запросов – свое описание выбираемых полей (ВЫБРАТЬ), источников запроса (ИЗ), условий отбора (ГДЕ), полей группировки (СГРУППИРОВАТЬ ПО). Затем данные, получаемые в результате каждого запроса, объединяются, и уже над этим объединением выполняются такие операции, как упорядочивание результатов (УПОРЯДОЧИТЬ ПО) и расчет итогов (ИТОГИ ПО).

ВЫБРАТЬ

Заказ.Ссылка.Клиент КАК Клиент,

Заказ.Товар КАК Товар,

СУММА(Заказ.Количество) КАК Заказано,

СУММА(0) КАК Продано

ИЗ

Документ.ЗаказТовара.Состав КАК Заказ

СГРУППИРОВАТЬ ПО

Заказ.Ссылка.Клиент,

Заказ.Товар

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

Накладная .Ссылка .Покупатель ,

Накладная .Товар ,

СУММА (0) ,

СУММА (Накладная .Количество)

ИЗ

Документ .РасходнаяНакладная .Состав КАК Накладная

СГРУППИРОВАТЬ ПО

Накладная .Ссылка .Покупатель ,

Накладная .Товар

ИТОГИ ПО

ОБЩИЕ ,

Клиент

По умолчанию при объединении запросов полностью одинаковые строки в результате запроса, сформированные разными запросами, заменяются одной. Если требуется, чтобы были оставлены разные строки, необходимо указать ключевое слово ВСЕ. В общем случае следует использовать конструкцию ОБЪЕДИНИТЬ ВСЕ.

Пакетные запросы и временные таблицы.

Для работы с временными таблицами нужен менеджер временных таблиц. Однако, удалось только в пределах одного запроса с формы на сервер. [Интересная статья](#) Глобальные таблицы 1С не использует.

```
&НаСервереБезКонтекста
```

```
Процедура СоздатьВременнуюТаблицуНаСервере()
```

```
□МенВТ = Новый МенеджерВременныхТаблиц;
```

```
□Запрос = Новый Запрос;
```

```
□Запрос.МенеджерВременныхТаблиц = МенВТ;
```

```
□Запрос.Текст ="
```

```
□□|ВЫБРАТЬ
```

```
□□| Поступление.Товар КАК Товар,
```

```
□□| Поступление.Цена КАК Цена,
```

```
□□| Поступление.Количество КАК Количество,
```

```
□□| Поступление.Сумма КАК Сумма
```

```
□□|ПОМЕСТИТЬ МоеПоступлениеТоваров
```

```
□□|ИЗ
```

```
□□| Документ.ПоступлениеТоваров.ПереченьТоваров КАК Поступление
```

```
□□|";
```

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

Этот менеджер будет актуален только до возврата из вызванной процедуры.

При Пакетном запросе менеджер создавать не нужно.

Временная таблица не существует в базе данных. Это область в памяти, созданная на ограниченное время. Эта область создается и заполняется данными при выполнении запроса, содержащего ключевое слово ПОМЕСТИТЬ, после которого следует произвольное имя временной таблицы.

```
ВЫБРАТЬ  
    Поступление.Товар,  
    Поступление.Ссылка.Дата КАК Дата,  
    Поступление.Ссылка.Поставщик  
ПОМЕСТИТЬ ПоступлениеТоваров  
ИЗ  
    Документ.ПриходнаяНакладная.Состав КАК Поступление  
ГДЕ  
    МЕСЯЦ(Поступление.Ссылка.Дата) = 11
```

Для быстрой и эффективной выборки данных из временной таблицы желательно проиндексировать эту таблицу по полям, которые будут затем участвовать в условии отбора или в условии соединения. Делается это с помощью предложения ИНДЕКСИРОВАТЬ ПО.

Пакетный запрос содержит последовательность запросов, разделенных символом «;» (точка с запятой). При выполнении пакетного запроса входящие в него запросы выполняются друг за другом.

```
ВЫБРАТЬ  
    Поступление.Товар,  
    Поступление.Ссылка.Дата КАК Дата,  
    Поступление.Ссылка.Поставщик  
ПОМЕСТИТЬ ПоступлениеТоваров  
ИЗ  
    Документ.ПриходнаяНакладная.Состав КАК Поступление  
ГДЕ  
    МЕСЯЦ(Поступление.Ссылка.Дата) = 11  
;  
ВЫБРАТЬ
```

Товары.Наименование,
Товары.Производитель,
ПоступлениеТоваров.Дата,
ПоступлениеТоваров.Поставщик

ИЗ

Справочник.Товары КАК Товары

ЛЕВОЕ СОЕДИНЕНИЕ ПоступлениеТоваров КАК ПоступлениеТоваров

ПО Товары.Ссылка = ПоступлениеТоваров.Товар

УПОРЯДОЧИТЬ ПО

Товары.Наименование ИЕРАРХИЯ

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

Общая информация.

Запуск конструктора: ПКМ в тексте модуля - Конструктор запроса или Основное меню Текст - Конструктор запроса.

Если вызвать конструктор при размещении курсора внутри существующего запроса, запрос будет проанализирован и либо перенесен в конструктор, либо будут указаны ошибки и конструктор не запустится до ручного исправления. Используется для проверки запроса, созданного вручную.

Кнопка Запрос в левом нижнем углу отображает текущий запрос без завершения работы из конструктора.

Вкладки

Без входа во вкладку, автоматические элементы в пределах функции вкладки не создаются.

Пакет запросов. Настройка состава пакета запросов. По умолчанию один запрос. При количестве вкладок больше одной справа появляются вертикальные вкладки для каждого запроса.

Таблицы и поля. Указание нужных таблиц и их полей в пределах одного запроса. В разделе Таблицы создается вложенный запрос кнопкой "Создать вложенный запрос", для использования полей временной таблицы нужно создать описание временной таблицы, вручную указав название таблицы, поля и тип данных. Для создания псевдонима таблицы (обычной, вложенной, виртуальной), в разделе таблицы ПКМ на имени таблицы - Переименовать таблицу.

Объединения/псевдонимы. В разделе справа в столбце Имя поля задать нужный псевдоним. В разделе слева можно убрать дубли. Для объединения, нужно в левом столбце добавить запрос. В столбце справа будет отображаться итоговая структура таблицы, но несопоставляемые поля будут заменяться NULL. Чтобы установить нужное значение (например 0), во вкладке Таблицы и поля нужно добавить произвольное поле.

Порядок. Для нужных полей устанавливаются правила, добавляется Автоупорядочивание.

Условия. Произвольное условие заполняется вручную, непроизвольное - только сравнение указанных столбцов таблиц. Условия по И.

Связи. Если в Таблицы и поля добавлены 2+ таблицы, то это поле доступно. Без галочек Все - внутреннее соединение. Одна галочка Все создает левое соединение (если одна галочка для правой таблицы, то все равно левое, но порядок меняется). Обе галочки создают полное

соединение.

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

Способы обхода выборки: в линейном порядке, в иерархическом порядке или по группировкам. Тип обхода выборки задается значением системного перечисления ОбходРезультатаЗапроса и передается в качестве параметра в метод Выбрать() объекта РезультатЗапроса. При линейном обходе выборки будут последовательно получаться все записи из результата запроса, при иерархическом обходе получают только записи результата запроса, находящиеся на одном уровне иерархии, при обходе по группировкам будут получены только родительские записи, являющие групповыми итогами.

Тип выборки	Описание	Функция в коде
Линейный	все записи из результата запроса	Без указания способа или СпособВыборки = ОбходРезультатаЗапроса.Прямой;
Иерархический	только записи результата запроса, находящиеся на одном уровне иерархии	СпособВыборки = ОбходРезультатаЗапроса.ПоГруппи ровкамСИерархией;
По группировкам	только родительские записи, являющие групповыми итогами	СпособВыборки = ОбходРезультатаЗапроса.ПоГруппи ровкам;

```
Запрос = Новый Запрос;  
Запрос.Текст = "";  
РезультатЗапроса = Запрос.Выполнить();  
  
СпособВыборки = ОбходРезультатаЗапроса.Прямой;  
ВыборкаЗапроса = РезультатЗапроса.Выбрать(СпособВыборки);  
  
Сообщение = Новый СообщениеПользователю;  
Пока ВыборкаЗапроса.Следующий() Цикл  
    Сообщение.Текст = "Товар: " + ВыборкаЗапроса.Товар.Наименование +  
        " Итого: " + ВыборкаЗапроса.Количество +  
        " Тип записи: " + ВыборкаЗапроса.ТипЗаписи() +  
        " Уровень: " + ВыборкаЗапроса.Уровень() +  
        " Группировка: " + ВыборкаЗапроса.Группировка();  
Сообщение.Сообщить();
```

```

&НаСервереБезКонтекста
Процедура ИерархическийОбходНаСервере()
□Запрос = Новый Запрос;
□Запрос.Текст =
□□"ВЫБРАТЬ
□□|□Товары.Товар КАК Товар,
□□|□Товары.Количество КАК Количество
□□|ИЗ
□□|□Документ.ПоступлениеТоваров.ПереченьТоваров КАК Товары
□□|
□□|УПОРЯДОЧИТЬ ПО
□□|□Товар
□□|ИТОГИ
□□| СУММА(Количество)
□□|ПО
□□| Товар Иерархия
□□|";
□РезультатЗапроса = Запрос.Выполнить();
□СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
□ВыборкаЗапроса = РезультатЗапроса.Выбрать(СпособВыборки);
□Сообщение = Новый СообщениеПользователю;
    Пока ИерархическаяВыборка.Следующий() Цикл
        Сообщение.Текст = "Товар: " + ИерархическаяВыборка.Товар.Наименование +
            " Количество: " + ИерархическаяВыборка.Количество +
            " Тип записи: " + ИерархическаяВыборка.ТипЗаписи() +
            " Уровень: " + ИерархическаяВыборка.Уровень() +
            " Группировка: " + ИерархическаяВыборка.Группировка();
        Сообщение.Сообщить();
    КонецЦикла;
КонецПроцедуры

```

Выгрузка в таблицу значений

```

Функция ПолучитьТекущиеЗадачиБизнесПроцесса(НазваниеЗадачиБП, СсылкаНаБизнесПроцесс) Экспорт
□// Возвращает список задач
□Запрос = Новый Запрос;
□Запрос.Текст =
    □□"ВЫБРАТЬ

```

```
□□|Задача.Ссылка КАК ТекущаяЗадача
□□|ИЗ
□□|Задача.ЗадачаМониторингаИС КАК Задача
□□|ГДЕ
□□|(НЕ Задача.Выполнена)
□□|И Задача.БизнесПроцесс = &ГлавныйБП
□□|И Задача.БизнесПроцесс.Стартован";
□Запрос.УстановитьПараметр("ГлавныйБП", СсылкаНаБизнесПроцесс);
□РезультатЗапроса = Запрос.Выполнить();
□ТЗ = РезультатЗапроса.Выгрузить();
□Для Каждого СтрокаТЗ из ТЗ Цикл
□□Сообщить(СтрокаТЗ[0]);
□КонецЦикла;
□Возврат ТЗ;
КонецФункции
```