

# Вложенные запросы, соединение, объединение, пакетные запросы

## 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

---

Revision #3

Created 20 January 2025 14:56:50 by Admin

Updated 20 January 2025 17:19:07 by Admin