

Глава 8. Работа с запросами

8.1. Общая информация

Для формирования и выполнения запросов к таблицам базы данных в системе используется специальный объект **Запрос**. Запрос удобно использовать, когда необходимо получить сложную выборку данных, сгруппированную и отсортированную нужным образом. Одним из классических примеров его применения может служить сводка по состоянию регистра учета на определенный момент времени. Кроме того, механизм запросов позволяет легко получать информацию в различных временных разрезах.

8.2. Источники данных (таблицы) запросов

В качестве источников данных языка запросов выступают таблицы базы данных. Таблицы подразделяются на два основных класса: реальные и виртуальные.

Реальные таблицы хранятся в базе данных, то есть интерпретируются из реально существующей таблицы базы данных. В случае использования реальной таблицы могут присутствовать вычисляемые поля, значения которых вычисляются как функция нескольких реальных полей.

Виртуальные таблицы не хранятся в базе данных. При обращении к информации виртуальных таблиц система автоматически собирает информацию реальных таблиц базы данных для выполнения запроса. Виртуальная таблица может быть параметризована, то есть реальное наполнение виртуальной таблицы может определяться значениями параметров, фактические значения которых задаются в тексте запроса. Для каждой виртуальной таблицы определяется имя, которое используется в запросах для идентификации таблицы.

Имя таблицы может быть задано на английском и русском языках. Например, **Справочник.Товары**. Имена таблиц и полей не могут совпадать с ключевыми словами языка запросов.

Отдельный подкласс таблиц образуют так называемые объектные таблицы. В качестве объектной таблицы обязательно выступает реальная таблица базы данных. Смысловое отличие объектных таблиц от прочих следует из названия - объектные таблицы предназначены для хранения состояния объектов системы «1С:Предприятие», таких как справочники, документы и т. п. Каждой объектной таблице соответствует тип объектов системы «1С:Предприятие». Например, объектам типа **Справочник.Товары** соответствует одна таблица, объектам типа **Справочник.Контрагенты** - другая. Каждая отдельная запись объектной таблицы хранит состояние отдельного объекта соответствующего типа. В соответствии с этим у каждой объектной таблицы определено поле типа **Ссылка** на текущую запись. Кроме того, для объектных таблиц определен способ получения пользовательского представления объекта из содержимого полей записи.

Объектные таблицы могут быть также иерархическими. Для иерархических таблиц определяется специально выделенное поле **Родитель**, содержащее ссылку на запись, которой в соответствии с иерархией подчиняется текущая запись.

В качестве поля таблицы может фигурировать поле таблицы или вложенная таблица. Основное отличие обычного поля от вложенной таблицы состоит в том, что в рамках одной записи обычному полю соответствует одно-единственное значение, а вложенной таблице соответствует значение типа **РезультатЗапроса** с заранее заданным набором колонок. Примером вложенной таблицы может являться табличная часть документа или справочника.

В качестве типов значения полей таблиц может выступать значение типа **NULL**. Такие значения содержатся в полях записей таблиц, для которых данное поле не определено или не имеет смысла. Например, значения такого типа содержатся в записях, относящихся к группам справочника, по полям, для которых установлено, что они могут иметь значение только у элементов этого справочника.

8.3. Виртуальные и обычные поля

Обычное поле - поле, которое формируется из данных, которые хранятся в СУБД, при этом данные могут быть дополнительно обработаны средствами СУБД. Виртуальное поле - поле, которое формируется из данных, которые хранятся в СУБД, но данные дополнительно обработаны

средствами сервера «1С:Предприятия» (в том числе в файловом варианте). Следующие поля в таблицах запросов являются виртуальными:

- Поля представлений, полученных как с помощью поля **Представление** соответствующей таблицы, так и с помощью функции языка запросов.
- **ВерсияДанных**;
- **ИмяПредопределенныхДанных**;
- **МоментВремени**;
- **ТипЗначения** таблицы плана видов характеристик;
- **Вид** таблицы плана счетов;
- **Тип** таблицы журнала документов;
- **ВидДвижения** таблицы регистра бухгалтерии;
- **ВидДвижения** таблицы регистра накопления;
- любые поля со значениями следующих типов: **ХранилищеЗначения**, **УникальныйИдентификатор** и ссылка внешнего источника данных.

Виртуальные поля имеют следующие особенности:

- Не все виртуальные поля можно преобразовать «обратно» в значение, хранящееся в СУБД. Такими полями являются поля представления и поле **ТипЗначения** для таблицы плана видов характеристик.
- Не все виртуальные поля с одинаковым наименованием можно сравнивать между собой, если они получены из различных таблиц. Такими полями являются **ВерсияДанных**, **ИмяПредопределенныхДанных**, **МоментВремени**, **Тип** таблицы журнала документов.
- Если в поле запроса получается значение системного перечисления (ссылки на таблицу внешних источников данных или уникального идентификатора), и полученное значение не комбинируется с другими типами, то при использовании в запросе операции **Сгруппировать По** значения этого поля группируются вне зависимости, от того, из каких таблиц получается значение.
- Для виртуальных полей строкового типа не поддерживается использование операций работы со строками и сравнение с данными типа **Строка**.

8.4. Язык запросов

8.4.1. Общая информация

Как было описано выше, для выполнения запроса необходимо составить текст запроса. Текст запроса - это инструкция, в соответствии с которой должен быть выполнен запрос. В тексте запроса описывается, какие таблицы информационной базы используются в качестве источников данных запроса, поля таблиц, которые требуется обрабатывать в запросе, правила группировки, сортировки результатов и т. д.

Инструкция составляется на специальном языке (языке запросов) и состоит из отдельных частей - секций, предложений, слов, функций и комментариев. Далее в этой главе рассматривается назначение и способы использования всех конструкций языка запросов.

8.4.2. Синтаксическая диаграмма конструкций языка запросов

В данной главе синтаксис языка запросов описывается при помощи набора правил. Каждое правило описывает одну конструкцию языка.

Каждая конструкция языка может содержать в себе ключевые слова; разделители (точки, запятые, круглые скобки), в свою очередь, другие конструкции языка.

```
ЭТО_КЛЮЧЕВОЕ_СЛОВО  
<Это_конструкция_языка>, <Это_конструкция_языка>  
ЭТО_ФУНКЦИЯ ( <Это_конструкция_языка> )
```

В правилах, описывающих язык запросов, конструкции языка указываются в угловых скобках. Ключевые слова и названия функций описываются заглавными буквами.

Конструкции языка могут содержать необязательные элементы - ключевые слова и пр. В правилах, описывающих язык запросов, необязательные элементы заключены в квадратные скобки.

```
[ЭТО_НЕОБЯЗАТЕЛЬНОЕ_СЛОВО]  
[<Это_необязательная_конструкция>]
```

В некоторых случаях в конструкции языка может использоваться один из нескольких альтернативных элементов. Такие элементы в правилах перечисляются через вертикальную черту.

```
ЛИБО_ЭТО_СЛОВО | ЛИБО_ЭТО_СЛОВО  
<Либо_эта_конструкция> | <Либо_эта_конструкция>
```

Описания всех конструкций сопровождаются примерами, поясняющими порядок их использования в языке запросов.

8.4.3. Комментарии в языке запросов

Текст запроса может включать комментарии. Комментарием считается часть строки, начинающаяся с последовательности символов // и продолжающаяся до конца строки.

```
// Это комментарий
```

При выполнении запроса комментарии игнорируются.

ПРИМЕЧАНИЕ. Конструктор запросов удаляет комментарии из текста.

8.4.4. Использование предопределенных данных конфигурации

Текст запроса может содержать предопределенные данные конфигурации, такие как:

- значения перечислений,
- предопределенные данные:
 - справочников,
 - планов видов характеристик,
 - планов счетов,
 - планов видов расчетов,
- пустые ссылки,
- значения точек маршрута бизнес-процессов.

Также текст запроса может содержать значения системных перечислений, которые могут быть присвоены полям в таблицах базы данных: ВидДвиженияНакопления, ВидСчета и ВидДвиженияБухгалтерии.

Обращение в запросах к предопределенным данным конфигурации и значениям системных перечислений осуществляется с помощью литерала функционального типа.

```
Значение (<ПредставлениеЗначения>)
```

Для системных перечислений представление значения имеет вид:

<ИмяСистемногоПеречисления>.<Значение>

Допустимые имена системных перечислений приведены выше, с перечнем допустимых для каждого из них значений можно ознакомиться в его описании.

Для predetermined данных конфигурации представление значения имеет вид:

<ТипПредопределенногоЗначения>.<ИмяОбъектаМетаданных>.<Значение>

Тип predetermined значения может быть:

- Справочник (Catalog);
- ПланВидовХарактеристик (ChartOfCharacteristicTypes);
- ПланСчетов (ChartOfAccounts);
- ПланВидовРасчета (ChartOfCalculationTypes);
- Перечисление (Enum).

В качестве имени объекта метаданных указывается имя объекта метаданных, как оно задано в конфигураторе.

Для определенных в конфигурации перечислений значение указывается как имя соответствующего объекта метаданных типа ЗначениеПеречисления. Для всех остальных типов predetermined значений - как имя predetermined элемента данных, как оно указано в конфигураторе, или ПустаяСсылка (EmptyRef) для указания пустой ссылки.

Для точек маршрутов бизнес-процессов представление значения имеет вид:

БизнесПроцесс.<ИмяОбъектаМетаданных>.ТочкаМаршрута.<ИмяТочкиМаршрута>

Ниже приведено несколько фрагментов запросов, поясняющих использование predetermined данных в запросах.

ГДЕ Город = Значение (Справочник.Города.Москва)
 ГДЕ Город = Значение (Справочник.Города.ПустаяСсылка)
 ГДЕ ТипТовара = Значение (Перечисление.ВидыТоваров.Услуга)
 ГДЕ ВидДвижения = Значение (ВидДвиженияНакопления.Приход)
 ГДЕ ТочкаМаршрута = Значение (БизнесПроцесс.Согласование.ТочкаМаршрута.Согласие)

8.4.5. Двухязычное представление ключевых слов

Одной из существенных особенностей языка запросов системы «1С:Предприятие» является то, что, как и во встроенном языке, все ключевые слова имеют два варианта написания: на русском и английском языках. Далее в этой главе указываются русские варианты написания ключевых слов. Ниже в таблице приведены соответствия русского и английского вариантов написания ключевых слов языка запросов.

Русское написание	Английское написание
АВТОУПОРЯДОЧИВАНИЕ	AUTOORDER
БУЛЕВО	BOOLEAN
В	IN
ВНЕШНЕЕ	OUTER
ВНУТРЕННЕЕ	INNER
ВОЗР	ASC
ВСЕ	ALL
ВЫБОР	CASE

ВЫБРАТЬ	SELECT
ВЫРАЗИТЬ	CAST
ГДЕ	WHERE
ГОД	YEAR
ДАТА	DATE
ДАТАВРЕМЯ	DATETIME
ДЕКАДА	TENDAYS
ДЕНЬ	DAY
ДЕНЬГОДА	DAYOFTIME
ДЕНЬНЕДЕЛИ	WEEKDAY
ДЛЯ ИЗМЕНЕНИЯ	FOR UPDATE [OF]
ДОБАВИТЬКДАТЕ	DATEADD
ЕСТЬ	IS
ЕСТЬNULL	ISNULL
Значение	VALUE
И	AND
ИЕРАРХИЯ	HIERARCHY
ИЗ	FROM
ИЛИ	Or
ИМЕЮЩИЕ	HAVING
ИНАЧЕ	ELSE
ИНДЕКСИРОВАТЬ ПО	INDEX BY
Истина	TRUE
ИТОГИ ... ПО	TOTALS ... BY
КАК	AS
КВАРТАЛ	QUARTER
КОГДА	WHEN
КОЛИЧЕСТВО	COUNT
КОНЕЦПЕРИОДА	ENDOFFPERIOD
КОНЕЦ	END
ЛЕВОЕ	LEFT
Ложь	FALSE
МАКСИМУМ	MAX
МЕЖДУ	BETWEEN
МЕСЯЦ	MONTH
МИНИМУМ	MIN
МИНУТА	MINUTE
НАЧАЛОПЕРИОДА	BEGINOFFPERIOD
НЕ	Not
НЕДЕЛЯ	WEEK

НЕОПРЕДЕЛЕНО	UNDEFINED
ОБЩИЕ	OVERALL
ОБЪЕДИНИТЬ	UNION
ПЕРВЫЕ	TOP
ПЕРИОДАМИ	PERIODS
ПОДОБНО	LIKE
ПОЛНОЕ	FULL
ПОЛУГОДИЕ	HALFYEAR
ПОМЕСТИТЬ	INTO
ПРАВОО	RIGHT
ПРЕДСТАВЛЕНИЕ	PRESENTATION
ПУСТАЯТАБЛИЦА	EMPTYTABLE
РАЗЛИЧНЫЕ	DISTINCT
РАЗРЕШЕННЫЕ	ALLOWED
Сгруппировать По	GROUP BY
СЕКUNДА	SECOND
СОЕДИНЕНИЕ ... ПО	JOIN ... ON
СПЕЦСИМВОЛ	ESCAPE
ПОДСТРОКА	SUBSTRING
СЕКUNДА	SECOND
СРЕДНЕЕ	AVG
ССЫЛКА	REFS
СТРОКА	STRING
СУММА	SUM
ТИП	TYPE
ТИПЗНАЧЕНИЯ	VALUETYPE
ТОГДА	THEN
ТОЛЬКО	ONLY
УБЫВ	DESC
УПОРЯДОЧИТЬ ПО	ORDER BY
ЧАС	HOURL
ЧИСЛО	NUMBER
УНИЧТОЖИТЬ	DROP

ПРИМЕЧАНИЕ. Регистр букв (строчные или заглавные) при написании не имеет значения.

8.4.6. Основные секции текста запроса

Текст запроса можно описать следующим правилом:

<Текст Запроса>

<Описание запроса>

[<Объединение запросов>]

[Копировать в буфер обмена](#)

[<Упорядочивание результатов>]
[АВТОУПОРЯДОЧИВАНИЕ]
[<Описание итогов>]

Как видно из этого правила, текст запроса состоит из нескольких частей, или секций:

- **<Описание запроса>** - это единственная обязательная секция в тексте запроса, и во многих случаях достаточно указать только ее. В секции определяются источники данных запроса, поля выборки, группировки и т. д. Эта секция, в свою очередь, описывается целым набором правил и подробно рассматривается ниже.
- **<Объединение запросов>** - язык запросов позволяет объединять результаты выполнения нескольких запросов. Описание объединения запросов см. [здесь](#).
- В секции **<Упорядочивание результатов>** можно определить условия упорядочивания строк в результате запроса. Описание упорядочивания результата запроса см. [здесь](#).
- Секция **АВТОУПОРЯДОЧИВАНИЕ** позволяет включить режим автоматического упорядочивания строк в результате запроса. Описание данного режима см. [здесь](#).
- В секции **<Описание итогов>** можно указать, какие итоги необходимо рассчитывать в запросе. Описание данной секции см. [здесь](#).

8.4.7. Описание запроса

8.4.7.1. Общая информация

Как уже было сказано, в тексте запроса должна обязательно присутствовать секция описания запроса, в которой определяются:

- поля, которые будут содержаться в результате запроса;
- источники данных запроса - исходные таблицы;
- условия, влияющие на выборку данных в запросе;
- порядок группировки результатов запроса.

Секция описания запроса состоит из нескольких взаимосвязанных предложений:

<Описание запроса>

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]  
    <Список полей выборки>  
[ИЗ <Список источников>]  
[ГДЕ <Условие отбора>]  
[Сгруппировать По <Поля группировки>]  
[ИМЕЮЩИЕ <Условие отбора>]  
[ДЛЯ ИЗМЕНЕНИЯ [<Список таблиц верхнего уровня>]]
```

Описание запроса начинается с обязательного ключевого слова **ВЫБРАТЬ**.

Ключевое слово **РАЗРЕШЕННЫЕ** означает, что запрос выберет только те записи, на которые у текущего пользователя есть права. Если данное слово не указать, то запрос отработает с ошибкой, когда он выберет записи, на которые у пользователя нет прав. Данное ключевое слово может присутствовать только в предложении **ВЫБРАТЬ** верхнего уровня и распространяется на весь запрос, включая вложенные запросы. Следует учитывать, что использование ключевого слова **РАЗРЕШЕННЫЕ** оказывает влияние на работу только в том случае, если на таблицы наложены ограничения доступа к данным (см. [здесь](#)). Права доступа на саму таблицу не учитываются при использовании ключевого слова **РАЗРЕШЕННЫЕ**. Например, если на таблицу отсутствует право **ЧТЕНИЕ**, то запрос с использованием такой таблицы, будет выполнен с ошибкой вне зависимости от использования в запросе ключевого слова **РАЗРЕШЕННЫЕ**.

С помощью ключевого слова **РАЗЛИЧНЫЕ** можно указать, что в результат не должны попадать повторяющиеся строки.

Конструкция **ПЕРВЫЕ** <Количество> позволяет задать предельное количество строк в результате запроса. Будут отобраны самые первые (в соответствии с правилами упорядочивания результатов запроса) строки. Количество задается целым числом. В языке запросов добавлена возможность исполнения упорядочивания во вложенных запросах в случае, если вложенный запрос содержит конструкцию **ПЕРВЫЕ**.

В секции <Список полей выборки> описываются поля, которые должны содержаться в результате запроса (правила описания полей выборки см. [здесь](#)).

В предложении **ИЗ** <Список источников> указываются источники данных - таблицы информационной базы, содержимое которых обрабатывается в запросе. Описание источников может быть опущено только в том случае, если они полностью определены в списке полей выборки (правила описания источников данных запроса см. [здесь](#)).

Предложение **ГДЕ** <Условие отбора> позволяет отфильтровать результат запроса. В результат попадают только те записи, для которых указанное условие оказывается истинным (правила описания условий отбора см. [здесь](#)).

Предложение **СТРУППИРОВАТЬ** позволяет описать порядок группировки результатов запроса (см. [здесь](#)).

Предложение **ИМЕЮЩИЕ** позволяет накладывать условия на результаты группировки (см. [здесь](#)).

Предложение **ДЛЯ ИЗМЕНЕНИЯ** предназначено для указания необходимости блокировки считываемых в транзакции данных. Считанные данные становятся недоступными для чтения в других сессиях. Для файлового варианта блокируются указанные таблицы, а для клиент-серверного варианта - только выбранные записи. Блокировка снимается после завершения транзакции.

Во всех примерах запросов в данной главе приводится текст запроса и результат запроса. Подразумевается, что текст запроса передается в качестве параметра методу **Выполнить ()** объекта **Запрос**.

Приведем пример достаточно простого запроса, состоящего из одного оператора **ВЫБРАТЬ** и списка полей выборки.

Пример:

[Копировать в буфер обмена](#)

```
// В отчет необходимо вывести список товарных накладных.
```

```
Выбрать
```

```
Документ.РасходнаяНакладная.Ссылка
```

Результат:

Ссылка
Расходная накладная 0000001 от 28.06.2006 14:19:00
Расходная накладная 0000002 от 28.06.2006 14:30:32
Расходная накладная 0000003 от 28.06.2006 14:30:49
Расходная накладная 0000004 от 28.06.2006 14:32:06
Расходная накладная 0000005 от 28.06.2006 14:32:32
Расходная накладная 0000006 от 28.06.2006 14:32:47
Расходная накладная 0000007 от 28.06.2006 14:34:04
Расходная накладная 0000008 от 28.06.2006 14:35:37
Расходная накладная 0000009 от 28.06.2006 14:36:05
Расходная накладная 0000010 от 28.06.2006 14:36:36
Расходная накладная 0000011 от 28.06.2006 14:37:04
Расходная накладная 0000012 от 28.06.2006 14:38:18
Расходная накладная 0000013 от 28.06.2006 14:45:29
Расходная накладная 0000014 от 28.06.2006 14:47:20
Расходная накладная 0000015 от 28.06.2006 14:48:09
Расходная накладная 0000016 от 28.06.2006 14:49:47
Расходная накладная 0000017 от 28.06.2006 14:50:23
Расходная накладная 0000018 от 28.06.2006 14:51:36
Расходная накладная 0000019 от 28.06.2006 14:58:16

Рис. 328. Результат запроса

8.4.7.2. Использование слова «РАЗЛИЧНЫЕ»

Во многих ситуациях желательно, чтобы одинаковые строки в отчете не повторялись.

Пример:

[Копировать в буфер обмена](#)

```
// Необходимо узнать, каким вообще контрагентам
// отгружался товар за период.
```

Выбрать

Документ.РасходнаяНакладная.Контрагент

Результат:

Контрагент
Эльбрус
Эльбрус
Алекс-2002
Большаков Андрей
Завод РТИ
Филипенко
Центр детского творчества
Алекс-2002
Магазин на ул. Алексеева
Никитин Юрий
Магазин на ул. Алексеева
Автохозяйство №34
Алекс-2002
Эльбрус
Русская одежда
Алекс-2002
Турмасов Марат Сергеевич
Завод РТИ
Магазин на ул. Алексеева

Рис. 329. Результат запроса

Видно, что в результате запроса много повторяющихся строк, что снижает его наглядность. Чтобы избежать повторений, в описании запроса следует указать ключевое слово **РАЗЛИЧНЫЕ**.

Пример:

[Копировать в буфер обмена](#)

Выбрать Различные
Документ.РасходнаяНакладная.Контрагент

Результат:

Контрагент
Эльбрус
Алекс-2002
Центр детского творчества
Магазин на ул. Алексеева
Автохозяйство №34
Завод РТИ
Русская одежда
Турмасов Марат Сергеевич
Большаков Андрей
Филипенко
Никитин Юрий

Рис. 330. Результат запроса

8.4.7.3. Использование слова «ПЕРВЫЕ»

В некоторых случаях необходимо вывести в отчет ограниченное количество строк. Для этого в описании запроса следует указать ключевое слово **ПЕРВЫЕ** и после него - требуемое количество строк.

Пример:

[Копировать в буфер обмена](#)

```
// Необходимо отобрать пять самых дорогих товаров.  
// Выборка должна осуществляться в порядке убывания цены товара.  
Выбрать Первые 5  
Справочник.Номенклатура.Наименование,  
Справочник.Номенклатура.ЗакупочнаяЦена  
Упорядочить По  
Справочник.Номенклатура.ЗакупочнаяЦена Убыв
```

Результат:

Наименование	ЗакупочнаяЦена
Копировальный аппарат Omega	3 500,00
Телефон Vega 700	2 500,00
Телефон Vega 300	2 200,00
Сист. блок Hewlett-Packard Vectra VL420	1 699,00
Сист. блок Hewlett-Packard Brio BA410	1 633,00

Рис. 331. Результат запроса

8.4.8. Описание полей выборки

После обязательного ключевого слова **ВЫБРАТЬ** (и уточняющих слов **РАЗЛИЧНЫЕ** и **ПЕРВЫЕ**) в тексте запроса задается список полей выборки. Эти поля будут обрабатываться при выборке данных в запросе. Результат запроса также будет иметь набор полей, определенный в данном списке. Поля выборки описываются по следующим правилам:

<Список полей выборки>

<Поле выборки>[, <Поле выборки>[, ...]] | *

<Поле выборки>

<Описание поля> | <Логическое выражение> [[КАК] <Псевдоним поля>]

<Описание поля>

<Выражение>[.<Группа полей>] | <Описание пустой таблицы>

<Описание пустой таблицы>

ПУСТАЯТАБЛИЦА. (<Список псевдонимов>)

<Список псевдонимов>

[<Псевдоним поля>] [, <Список псевдонимов>]

Список полей выборки состоит из одного или нескольких элементов, которые разделены запятыми. Каждое поле выборки состоит из описания поля выборки и необязательного псевдонима поля.

Вместо перечисления полей в списке выборки можно указать звездочку (*). Это будет означать, что результат запроса должен содержать все поля, которые есть в исходных таблицах - источниках данных запроса, описанных в списке источников.

Примечание 1. При указании звездочки (*) в списке полей выборки в результат не включаются виртуальные поля исходных таблиц.

Примечание 2. Получение выборок очень большого размера (более **64 МБ**) требует наличия достаточного количества свободного места на диске, используемом для размещения временных файлов сервера и клиента.

Описание поля определяет, каким образом должны формироваться значения поля. В простейшем случае поле выборки является ссылкой на поле исходной таблицы. Ссылка может задаваться с указанием таблицы, содержащей это поле, или без указания самой таблицы. Разыменование полей см. стр. [833](#).

В общем случае поле выборки может представлять собой не только ссылку на поле исходной таблицы, а некоторое выражение (подробнее см. [здесь](#)) или логическое выражение (см. [здесь](#)).

Результаты запроса могут быть сгруппированы (см. [здесь](#)) с помощью агрегатных функций (см. [здесь](#)), указанных в качестве выражений в полях выборки.

Каждому полю выборки может быть назначен псевдоним (см. [здесь](#)). В дальнейшем псевдоним поля может использоваться для более удобного обращения к данному полю.

Группа полей может указываться только тогда, когда поле выборки указывает на вложенную таблицу (см. [здесь](#)). В этом случае можно указать, какие поля должны обрабатываться в выборке по вложенной таблице. Если группа полей не указана, в выборке будут обрабатываться все поля вложенной таблицы.

8.4.8.1. Псевдонимы полей в списке выборки

Если полю выборки назначить псевдоним, то в дальнейшем к этому полю можно будет обращаться, используя его псевдоним, в предложениях **УПОРЯДОЧИТЬ ПО** и **ИТОГИ**, а также при работе с результатом запроса. Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным.

Ключевое слово **КАК** может предшествовать псевдониму поля. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса.

Псевдонимы полей задаются в соответствии с правилами назначения идентификаторов переменных. Псевдонимы в запросе не могут совпадать.

Назначение псевдонимов полям само по себе никак не влияет на выборку данных в запросе.

Пример:

[Копировать в буфер обмена](#)

```
// Необходимо выбрать из справочника товаров
// наименования товаров и наименования групп.
Выбрать
Справочник.Номенклатура.Наименование Как Товар,
Справочник.Номенклатура.Родитель.Наименование Как Группа
Из
Справочник.Номенклатура
```

Результат:

Товар	Группа
... Установка ПО	Услуги
... Консультации по настройке ОС Windows	Услуги
... Консультации по настройке 1С	Услуги
... Ноутбуки	
... Телефоны	
... Копировальные аппараты	
... Копировальный аппарат Omega	Копировальные аппараты
... Ноутбук Rover Computers Navigator KT7	Ноутбуки
... Ноутбук Rover Computers Explorer	Ноутбуки
... Телефон Vega 700	Телефоны
... Телефон Vega 300	Телефоны
... Телефон Siemens SL45	Телефоны
... Телефон I G W7200	Телефоны

Рис. 332. Результат запроса (фрагмент)

Следует обратить внимание, что поля в результате запроса называются **Товар** и **Группа**. Если бы псевдонимы полей не были указаны, поля в результате запроса назывались бы **Наименование** и **Наименование1** (названия полей в результате запроса не могут совпадать, поэтому к названию второго поля автоматически добавлено «1»), что менее наглядно.

8.4.8.2. Вложенные таблицы в списке полей выборки

Поле в списке выборки может ссылаться на вложенную таблицу источника данных запроса. В этом случае поле результата запроса будет иметь тип **РезультатЗапроса**, то есть содержать вложенный результат запроса, сформированный на основе вложенной таблицы - источника.

По умолчанию во вложенный результат включаются все поля вложенной таблицы - источника данных. Имеется возможность явно определить группу полей, которые должны содержаться во вложенном результате запроса. Группа полей вложенного результата описывается по следующему правилу:

<Группа полей>

[Копировать в буфер обмена](#)

```
( <Список вложенных полей> ) | *
```

<Список вложенных полей>

[Копировать в буфер обмена](#)

```
<Вложенное поле>[, <Вложенное поле>[, ...]]
```

<Вложенное поле>

[Копировать в буфер обмена](#)

```
<Выражение> [[КАК] <Псевдоним поля>]
```

Список вложенных полей состоит из одного или нескольких элементов, разделенных запятыми. Если список состоит из одного элемента, его не обязательно заключать в скобки.

Вместо перечисления вложенных полей можно указать звездочку (*); это будет означать, что результат запроса должен содержать все поля, которые есть во вложенной таблице.

Вложенное поле может представлять некоторое выражение (см. [здесь](#)). В простейшем случае выражение - это ссылка на поле вложенной таблицы.

Каждому вложенному полю может быть назначен псевдоним. В дальнейшем псевдоним поля может использоваться для более удобного обращения к данному полю, аналогично псевдонимам полей списка выборки - см. [здесь](#).

Псевдонимы вложенным полям могут быть назначены независимо от того, задан ли псевдоним самой вложенной таблице.

Пример:

[Копировать в буфер обмена](#)

```
// В отчет необходимо вывести спецификацию товарных накладных  
// сам документ, номенклатуру и количество.
```

Выбрать

```
Документ.РасходнаяНакладная.Ссылка,  
Документ.РасходнаяНакладная.Состав. (Номенклатура Как Товар, Количество)
```

Результат:

Ссылка	Состав
Расходная накладная 0000001 от 28.06.2006 14:19:00	ТаблицаЗначений
Расходная накладная 0000002 от 28.06.2006 14:30:32	ТаблицаЗначений
Расходная накладная 0000003 от 28.06.2006 14:30:49	ТаблицаЗначений
Расходная накладная 0000004 от 28.06.2006 14:30:59	ТаблицаЗначений
Расходная накладная 0000005 от 28.06.2006 14:31:09	ТаблицаЗначений
Расходная накладная 0000006 от 28.06.2006 14:31:19	ТаблицаЗначений
Расходная накладная 0000007 от 28.06.2006 14:31:29	ТаблицаЗначений
Расходная накладная 0000008 от 28.06.2006 14:31:39	ТаблицаЗначений
Расходная накладная 0000009 от 28.06.2006 14:31:49	ТаблицаЗначений
Расходная накладная 0000010 от 28.06.2006 14:31:59	ТаблицаЗначений
Расходная накладная 0000011 от 28.06.2006 14:32:09	ТаблицаЗначений
Расходная накладная 0000012 от 28.06.2006 14:32:19	ТаблицаЗначений
Расходная накладная 0000013 от 28.06.2006 14:45:29	ТаблицаЗначений
Расходная накладная 0000014 от 28.06.2006 14:47:20	ТаблицаЗначений
Расходная накладная 0000015 от 28.06.2006 14:48:09	ТаблицаЗначений
Расходная накладная 0000016 от 28.06.2006 14:49:47	ТаблицаЗначений

Вложенная таблица

Товар	Количество
1С:Бухгалтерия 8. Базовая версия	2,00
1С:Бухгалтерия 8.	2,00

Рис. 333. Результат запроса (фрагмент)

Следует обратить внимание, что поле **Состав** результата запроса представляет собой вложенную таблицу, имеющую поля **Номенклатура** и **Количество**.

8.4.8.3. Пустые вложенные таблицы в списке выборки

Если в запросе используется объединение и в некоторых частях объединения присутствуют вложенные таблицы, а в некоторых - нет, возникает необходимость дополнения списка выборки полями - пустыми вложенными таблицами. Делается это при помощи ключевого слова **ПУСТАЯТАБЛИЦА**. После него в скобках указываются псевдонимы полей, из которых будет состоять вложенная таблица.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ Ссылка.Номер, ПУСТАЯТАБЛИЦА. (Ном, Тов, Кол) КАК Состав  
ИЗ Документ.РасходнаяНакладная  
ОБЪЕДИНИТЬ ВСЕ  
ВЫБРАТЬ Ссылка.Номер, Состав. (НомерСтроки, Номенклатура, Количество)  
ИЗ Документ.РасходнаяНакладная
```

8.4.9. Описание источников запроса

Задача предложения **ИЗ** состоит в том, чтобы обозначить список исходных таблиц - источников данных, используемых в данном операторе **ВЫБРАТЬ**.

Следует отметить, что предложение **ИЗ** в языке запросов является опциональным. Оно может быть опущено в том случае, если источники данных полностью квалифицированы в описании списка полей выборки, содержащегося в предложении **ВЫБРАТЬ**. Следует обратить внимание, что ряд примеров в предыдущих разделах не содержал предложения **ИЗ**.

После ключевого слова **ИЗ** указывается список источников. В общем случае список источников описывается следующим набором правил:

<Список источников>

[Копировать в буфер обмена](#)

<Источник>[, <Источник>[, ...]]

<Источник>

[Копировать в буфер обмена](#)

<Описание источника> [<Перечень соединений>]

<Описание источника>

[Копировать в буфер обмена](#)

<Таблица> [[КАК] <Псевдоним источника>]

<Таблица>

[Копировать в буфер обмена](#)

<Имя таблицы>[(<Параметры>)] | (<Описание запроса>)

Источники данных запроса перечисляются в списке источников через запятую. Каждый источник в списке источников обязательно включает в себя описание источника; кроме того, может быть указан перечень соединений - правила соединений источника с другими источниками. Описание спецификации соединений см. [здесь](#).

Если в качестве источника данных выступает таблица информационной базы, описание источника содержит имя таблицы.

Если исходная таблица виртуальная, могут быть указаны параметры ее формирования. Подробно параметры виртуальных таблиц описаны в разделе [Встроенный язык - Работа с запросами - Таблицы запросов](#) встроенной справки.

В качестве источника данных запроса может выступать также вложенный запрос. В этом случае описание источника содержит описание запроса. Описание использования вложенных запросов см. [здесь](#).

В описании источника данных может быть также назначен его псевдоним (см. стр. [814](#)). В дальнейшем псевдоним источника может использоваться для более удобного обращения к данному источнику.

8.4.9.1. Спецификации соединений

8.4.9.1.1. Общая информация

При определении нескольких источников в списке для каждой записи из первой таблицы-источника осуществляется выборка из второй таблицы-источника и т. д. Таким образом, в результате запроса формируются все возможные комбинации всех записей из всех указанных источников.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
Контрагенты.Ссылка как Контрагент,
ТипыЦен.Ссылка КАК ТипыЦен
ИЗ
Справочник.Контрагенты КАК Контрагенты,
Справочник.ТипыЦен КАК ТипыЦен
```

Результат:

Контрагент	ТипЦены
Темп плюс	Оптовая
Темп плюс	Закупочная
Темп плюс	Розничная
Темп плюс	Мелкооптовая
Поставщики	Оптовая
Поставщики	Закупочная
Поставщики	Розничная
Поставщики	Мелкооптовая
Максимус	Оптовая
Максимус	Закупочная
Максимус	Розничная
Максимус	Мелкооптовая
Новация	Оптовая
Новация	Закупочная
Новация	Розничная
Новация	Мелкооптовая

Рис. 334. Результат запроса (фрагмент)

Результат запроса содержит комбинации всех контрагентов со всеми типами цен. Как правило, такой результат сам по себе смысла не имеет. Обычно комбинации записей из разных исходных таблиц требуется ограничить какими-либо условиями. В языке запросов имеется возможность описать такое соединение источников, указывая сами источники и определяя условия, в соответствии с которыми комбинации записей из этих источников требуется включить в результат запроса.

Соединения бывают нескольких видов; они описываются следующими правилами:

<Перечень соединений>

[Копировать в буфер обмена](#)

<Соединение> [<Перечень соединений>]

<Соединение>

[Копировать в буфер обмена](#)

[ВНУТРЕННЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
 ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
 ПРАВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
 ПОЛНОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора>

В общем случае перечень соединений может содержать и описывать не только одно соединение (двух источников), но и несколько соединений нескольких источников сразу.

Описание источника содержит описание исходной таблицы (см. [здесь](#)).

Условие отбора содержит условия, в соответствии с которыми в выборке необходимо соединить данные из исходных таблиц - источников запроса. Правила описания условий в языке запросов см. [здесь](#).

Ключевые слова ЛЕВОЕ, ПРАВОЕ и ПОЛНОЕ уточняют характер соединения. Слова ВНУТРЕННЕЕ или ВНЕШНЕЕ можно не указывать вообще, они повышают наглядность и удобочитаемость текста запроса.

Соединяемые источники не равнозначны между собой, и в некоторых случаях результат зависит от того, какая таблица указана первой, до ключевого слова СОЕДИНЕНИЕ (слева от него), а какая - второй (справа).

Для описания соединений будем использовать две простые таблицы. Одна таблица называется Компании, состоит из двух полей: Наименование и Телефон и содержит следующие данные:

Наименование	Телефон
Конфетпром	
СервисПог	2-12-85-03

Рис. 335. Таблица «Компании»

Другая таблица называется **Контакты**, состоит из трех полей: **Наименование**, **Телефон** и **Компания** (ссылка на элемент таблицы **Компании**). Таблица содержит следующие данные:

Наименование	Телефон	Компания
Абдулов Юрий Владимирович	222-3-22	
Семенов Геннадий Сергеевич		СервисПог
Шилов Сергей Александрович	222-3-22	Конфетпром

Рис. 336. Таблица «Контакты»

ПРИМЕЧАНИЕ. В терминах «1С:Предприятия» обе таблицы являются справочниками.

8.4.9.1.2. Внутреннее соединение

Внутреннее соединение означает, что из обеих исходных таблиц - источников данных в результат запроса необходимо включить только те комбинации записей, которые соответствуют указанному условию. Остальные записи в результат не попадают.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Компании.Наименование КАК Компания,
Контакты.Наименование КАК Контакт

ИЗ

Справочник.Контакты КАК Контакты
ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
ПО Контакты.Компания = Компании.Ссылка

Результат:

Компания	Контакт
Конфетпром	Шилов Сергей Александрович
СервисПог	Семенов Геннадий Сергеевич

Рис. 337. Результат запроса

8.4.9.1.3. Левое внешнее соединение

Левое внешнее соединение означает, что в результат запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Но, в отличие от внутреннего соединения, в результат запроса нужно включить также еще и записи из первого (указанного слева от слова **СОЕДИНЕНИЕ**) источника, для которых не найдено соответствующих условию записей из второго источника.

Таким образом, в результат запроса будут включены все записи из первого источника; они будут соединены с записями из второго источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из второго источника, будут содержать **NULL** в полях, формируемых на основании записей из этого источника.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Контакты.Наименование КАК Контакт,
Компании.Наименование КАК Компания

ИЗ

Справочник.Контакты КАК Контакты
ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
ПО Контакты.Компания = Компании.Ссылка

Результат:

Контакт	Компания
Шилов Сергей Александрович	Конфетпром
Абдулов Юрий Владимирович	NULL
Семенов Геннадий Сергеевич	СервисПог

Рис. 338. Результат запроса

8.4.9.1.4. Правое внешнее соединение

Правое внешнее соединение означает, что в результат запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса нужно также включить и записи из второго (указанного справа от слова **СОЕДИНЕНИЕ**) источника, для которых не найдено соответствующих условию записей из первого источника.

Таким образом, в результат запроса будут включены все записи из второго источника; они будут соединены с записями из первого источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из первого источника, будут содержать **NULL** в полях, формируемых на основании записей из этого источника.

ПРИМЕЧАНИЕ. Фактически правое внешнее соединение можно выразить через левое внешнее соединение. Именно так делает конструктор запросов конфигулятора.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Контакты.Наименование КАК Контакт,
    Компании.Наименование КАК Компания
ИЗ
    Справочник.Контакты КАК Контакты
    ПРАВОЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
    ПО Контакты.Телефон = Компании.Телефон

```

Результат:

Контакт	Компания
NULL	СервисПог
Семенов Геннадий Сергеевич	Конфетпром

Рис. 339. Результат запроса

8.4.9.1.5. Полное внешнее соединение

Полное внешнее соединение означает, что в результат запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса нужно также включить и те записи из обоих источников, для которых не найдено соответствий.

Таким образом, в результат запроса будут включены все записи из обоих источников; они будут соединены друг с другом при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из какого-либо источника, будут содержать **NULL** в полях, формируемых на основании записей из этого источника.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Контакты.Наименование КАК Контакт,
    Компании.Наименование КАК Компания
ИЗ
    Справочник.Компании КАК Компании
    ПОЛНОЕ СОЕДИНЕНИЕ Справочник.Контакты КАК Контакты
    ПО (Контакты.Телефон = Компании.Телефон)

```

Результат:

Контакт	Компания
NULL	СервисПог
Семенов Геннадий Сергеевич	Конфетпром
Шилов Сергей Александрович	NULL
Абдулов Юрий Владимирович	NULL

Рис. 340. Результат запроса

8.4.9.2. Псевдонимы источников данных

Если источнику данных назначить псевдоним, то в дальнейшем к этому источнику можно будет обращаться, используя этот псевдоним (и уже нельзя будет обращаться через указание имени таблицы). Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным.

Псевдоним задается в соответствии с правилами назначения идентификаторов переменных. Псевдонимы в запросе не могут совпадать.

Ключевое слово **КАК** может предшествовать псевдониму источника. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса.

Назначение псевдонимов источникам само по себе никак не влияет на выборку данных в запросе.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Товар.Наименование,
    Товар.Родитель
ИЗ
    Справочник.Номенклатура КАК Товар
```

Данный пример демонстрирует использование в списке полей выборки псевдонима **Товар**, назначенного исходной таблице **Справочник.Номенклатура**.

8.4.9.3. Вложенные таблицы в списке источников

В списке источников могут фигурировать и вложенные таблицы - табличные части справочников и документов.

Пример:

Копировать в буфер обмена

```
// В отчет необходимо вывести спецификацию товарных накладных
// показать сам документ, номенклатуру и количество.
// В списке источников указана вложенная таблица "Состав"
// табличная часть накладной.
// Выборка ограничена восемью записями, чтобы не перегружать пример.
ВЫБРАТЬ ПЕРВЫЕ 8
    РасходнаяНакладнаяСостав.Ссылка,
    РасходнаяНакладнаяСостав.Номенклатура,
    РасходнаяНакладнаяСостав.Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
```

Результат:

Ссылка	Номенклатура	Количество
Расходная накладная 0000001 от 28.06.2006 14:19:00	1С:Бухгалтерия 8. Базовая версия	2,00
Расходная накладная 0000001 от 28.06.2006 14:19:00	1С:Бухгалтерия 8.	2,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Монитор 15" LG Studioworks 575N	5,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь 2-кноп A4Tech PS/2	10,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь LOGITECH M-S48 PS/2	10,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь OK-720 Mouse A4Tech PS/2	2,00
Расходная накладная 0000003 от 28.06.2006 14:30:49	1С:Зарплата и Управление Персоналом 8	1,00
Расходная накладная 0000004 от 28.06.2006 14:32:06	Windows XP Home Edition Russian CD	1,00

Рис. 341. Результат запроса

Следует обратить внимание, что при указании вложенной таблицы в списке источников допускается обращение как к полям самой вложенной таблицы, так и к полям таблицы верхнего уровня (той, которая содержит вложенную таблицу) через поле `Ссылка`. В данном случае происходит обращение к полю `Ссылка.НаименованиеПоля` самого документа.

8.4.9.4. Вложенные запросы в списке источников

В списке источников запроса в качестве таблицы-источника может использоваться вложенный запрос. В этом случае описание источника содержит описание вложенного запроса. Описание вложенного запроса составляется точно так же, как и обычного (см. [здесь](#)).

Использование вложенного запроса в качестве источника данных ничем не отличается от использования таблицы информационной базы. В качестве полей такого источника доступны все поля, описанные в списке полей выборки вложенного запроса.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

ВложенныйЗапрос.Ссылка,
ВложенныйЗапрос.Номенклатура,
ВложенныйЗапрос.Количество

ИЗ

(**ВЫБРАТЬ ПЕРВЫЕ 8**

РасходнаяНакладнаяСостав.Ссылка **КАК** Ссылка,
РасходнаяНакладнаяСостав.Номенклатура **КАК** Номенклатура,
РасходнаяНакладнаяСостав.Количество **КАК** Количество

ИЗ

Документ.РасходнаяНакладная.Состав **КАК** РасходнаяНакладнаяСостав) **КАК** ВложенныйЗапрос

Результат будет точно таким же, как и в предыдущем примере.

8.4.10. Фильтрация результатов запроса

Предложение **ГДЕ** <Условие отбора> позволяет задать условие отбора данных из исходных таблиц - источников запроса; в запросе будут обрабатываться только те записи, для которых данное условие оказывается истинным.

Пример:

[Копировать в буфер обмена](#)

// Необходимо выяснить, какие контрагенты являются частными лицами.

ВЫБРАТЬ

Контрагенты.Наименование

ИЗ

Справочник.Контрагенты **КАК** Контрагенты

ГДЕ Контрагенты.Вид = Значение (Перечисление.ВидыКонтрагентов.ЧастноеЛицо)

Результат:

Наименование
Эльбрус
Турмасов Марат Сергеевич
Большаков Андрей
Филипенко
Пугачев Максим Олегович
Никитин Юрий

Рис. 342. Результат запроса

ПРИМЕЧАНИЕ. Совершенно не обязательно, чтобы поле, фигурирующее в предложении **ГДЕ**, входило в список выборки.

Условие отбора может определяться и как простое логическое выражение, и как более сложное, в котором простые логические выражения соединяются между собой логическими операторами **И**, **Или**, **Не**. Описание условий в языке запросов см. [здесь](#).

При выборке данных из таблицы, имеющей табличные части, условия на записи табличной части действуют как на состав записей, выбираемых из таблицы, так и на состав записей табличных частей.

В общем виде: выборка записей, содержащих табличные части, выполняется в следующем порядке:

- К каждой таблице из списка источников левым внешним соединением присоединяются ее табличные части.
- Выполняются все соединения, определенные в списке источников.
- Отбираются записи, удовлетворяющие условию из раздела **ГДЕ**.
- Выполняется группировка записей по полям таблиц из списка источников, включая поля ссылок, с помещением записей табличных частей в качестве значений соответствующих полей записей таблиц из списка выборки.

8.4.11. Группировка результатов запроса

Исходные данные в запросе могут быть сгруппированы с помощью агрегатных функций, указанных в качестве полей в списке выборки. Это означает, что строки в результате запроса будут содержать результаты вычисления указанных агрегатных функций, рассчитанные (сгруппированные) по записям исходных таблиц.

Сами агрегатные функции указываются в списке полей выборки. В предложении **Сгруппировать По <Поля группировки>** необходимо указать список полей, по которым следует произвести группировку. В запросе будут группироваться записи исходных таблиц, содержащие одинаковые значения указанных полей.

Список полей группировки содержит ссылки на поля исходных таблиц - источников запроса, указанные через запятую:

<Поля группировки>

<Разыменование поля> [, <Разыменование поля> [, ...]]

ВНИМАНИЕ! При группировке результатов запроса в списке полей выборки обязательно должны быть указаны агрегатные функции, а помимо них допускается указывать только поля, по которым осуществляется группировка. В запросах, содержащих группировку по какому-либо полю, не следует использовать это поле внутри агрегатных функций. Использование поля одновременно для группировки и внутри агрегатной функции - следует считать ошибкой. Такие запросы могут трактоваться по-разному, в зависимости от того, какая СУБД используется. При нарушении данного правила возможны ошибки, приводящие к завершению программы.

Исключение составляют ситуации, когда агрегатные функции применены к полям вложенной таблицы. В этом случае в списке полей выборки возможны обращения к полям таблицы верхнего уровня, без группировки результатов по этим полям.

При использовании агрегатных функций предложение **Сгруппировать По** может быть и не указано совсем; при этом все результаты запроса будут сгруппированы в одну-единственную строку.

Пример:

[Копировать в буфер обмена](#)

```
// Требуется получить статистику по продаже товаров:
// максимальную, минимальную и среднюю цены в расходных накладных.
```

Выбрать

```
Накладная.Номенклатура,
Среднее (Накладная.Цена) Как Среднее,
Максимум (Накладная.Цена) Как Максимум,
Минимум (Накладная.Цена) Как Минимум
```

Из

```
Документ.РасходнаяНакладная.Состав Как Накладная
```

Сгруппировать По

```
Накладная.Номенклатура
```

Результат:

Номенклатура	Среднее	Максимум	Минимум
Монитор 15' LG Studioworks 575N	151,666667	155,00	145,00
Монитор 17' Philips 107S20	210	210,00	210,00
Монитор 19' Hitachi CM715ET	350	350,00	350,00
Монитор LCD 22' M8537ZM/A	2 000	2 000,00	2 000,00
Сист. блок Hewlett-Packard Brio BA410	2 000	2 000,00	2 000,00
Сист. блок Hewlett-Packard Vectra VL420	2 100	2 100,00	2 100,00
Сист. блок IBM NetVista A22p	1 816,666667	2 500,00	1 350,00
Сист. блок IBM NetVista M41	1 450	1 450,00	1 450,00
Лазерный принтер Canon I RP-R10	300	300,00	300,00

Рис. 343. Результат запроса (фрагмент)

8.4.12. Условия на значения агрегатных функций

Предложение **ИМЕЮЩИЕ <Условие отбора>** позволяет накладывать условия на значения агрегатных функций. В других конструкциях языка запросов, например, в предложении **ГДЕ**, указывать в условиях агрегатные функции нельзя.

Пример:

[Копировать в буфер обмена](#)

```
// Необходимо выбрать товары, которых продали более 20 штук.
```

ВЫБРАТЬ

```
Накладная.Номенклатура,
СУММА (Накладная.Количество) КАК Количество
```

ИЗ

```
Документ.РасходнаяНакладная.Состав КАК Накладная
```

Сгруппировать По

```
Накладная.Номенклатура
```

ИМЕЮЩИЕ

```
СУММА (Накладная.Количество) > 20
```

Результат:

Номенклатура	Количество
Лазерный принтер HP LaserJet 2200	26
Мышь 2-кноп A4Tech PS/2	35
Мышь Ice Mouse MUS-2	30
Мышь LOGITECH M-S48 PS/2	41
Мышь GENIUS "EASY" (3 кнопки).	21
Клавиатура Apple Pro Keyboards	23
Клавиатура LK-601 KB-2000 PS/2	23

Рис. 344. Результат запроса

ВНИМАНИЕ! В условии отбора можно использовать только агрегатные функции и поля, по которым осуществляется группировка.

8.4.13. Объединение запросов

В языке запросов имеется возможность объединять несколько запросов. При этом записи, полученные с помощью каждого из объединяемых запросов, будут собраны в один результат запроса.

При объединении каждый запрос собирает данные независимо, а такие операции, как упорядочивание результатов и расчет итогов, выполняются уже над результатом объединения запросов.

Поля результата запроса будут называться так, как описано в списке полей выборки первого из объединяемых запросов. Поля выборки остальных запросов сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки. Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки.

Если поля выборки объединяемых запросов имеют разный тип, то поля результата запроса будут иметь составной тип.

Объединение запросов описывается по следующему правилу:

<Объединение запросов>

[Копировать в буфер обмена](#)

```
ОБЪЕДИНИТЬ [ВСЕ]  
<Описание запроса>  
[<Объединение запросов>]
```

Объединение запросов начинается с обязательного ключевого слова **ОБЪЕДИНИТЬ**, после которого следует описание присоединяемого запроса. Далее может присоединяться еще один запрос и т. д.

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**.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ  
    Приход.Номенклатура КАК Товар,  
    СУММА (Приход.Количество) КАК Приход,  
    СУММА (0) КАК Расход  
ИЗ  
    Документ.ПриходнаяНакладная.Состав КАК Приход  
Сгруппировать По  
    Приход.Номенклатура  
ОБЪЕДИНИТЬ  
ВЫБРАТЬ  
    Расход.Номенклатура,  
    СУММА (0) ,  
    СУММА (Расход.Количество)  
ИЗ  
    Документ.РасходнаяНакладная.Состав КАК Расход  
Сгруппировать По  
    Расход.Номенклатура
```

Результат:

Товар	Приход	Расход
Клавиатура LK-601 KB-2000 PS/2	160	
Windows XP Home Edition Russian CD		3
Windows XP Home Edition Russian CD	25	
Windows XP Home Edition Russian UPG CD		3
Windows XP Home Edition Russian UPG CD	42	
Windows XP Professional Russian CD		6
Windows XP Professional Russian CD	15	
1С:Бухгалтерия 8. Базовая версия		2
1С:Бухгалтерия 8. Базовая версия	16	
1С:Бухгалтерия 8.		2

Рис. 345. Результат запроса (фрагмент)

8.4.14. Упорядочивание результатов запроса

Предложение **УПОРЯДОЧИТЬ ПО** позволяет сортировать строки в результате запроса.

<Упорядочивание результатов>

[Копировать в буфер обмена](#)

УПОРЯДОЧИТЬ ПО <Условия упорядочивания>

<Условия упорядочивания>

[Копировать в буфер обмена](#)

<Поле упорядочивания> [<Порядок>]
[, <Поле упорядочивания> [<Порядок>][, ...]]

<Поле упорядочивания>

[Копировать в буфер обмена](#)

<Выражение> <Порядок> ВОЗР | УБЫВ | ИЕРАРХИЯ | ИЕРАРХИЯ УБЫВ

В предложении **УПОРЯДОЧИТЬ ПО** через запятую перечисляются условия, в соответствии с которыми необходимо упорядочить результат запроса. Выборки упорядочиваются сначала по первому условию, потом по второму и т. д.

Условие упорядочивания в общем случае может представлять собой некоторое выражение (см. [здесь](#)). Строки результата запроса будут упорядочены по значениям этого выражения, рассчитанным для каждой строки.

Упорядочивание может осуществляться в порядке возрастания или убывания значений, а для таблиц, для которых задано свойство иерархичности, - также и по иерархии (см. [здесь](#)). Порядок может задаваться для каждого поля независимо. Описание правил сравнения значений см. [здесь](#).

Поле, фигурирующее в условиях упорядочивания, совсем не обязательно должно попадать в результат запроса.

Пример:

[Копировать в буфер обмена](#)

```
// Требуется отобрать 5 самых дорогих товаров,
// расположив их в порядке убывания цены.
ВЫБРАТЬ ПЕРВЫЕ 5
    Номенклатура.Наименование,
    Номенклатура.ЗакупочнаяЦена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
УПОРЯДОЧИТЬ ПО
    Номенклатура.ЗакупочнаяЦена УБЫВ
```

Результат:

Наименование	ЗакупочнаяЦена
Копировальный аппарат Omega	3 500,00
Телефон Vega 700	2 500,00
Телефон Vega 300	2 200,00
Сист. блок Hewlett-Packard Vectra VL420	1 699,00
Сист. блок Hewlett-Packard Brio BA410	1 633,00

Рис. 346. Результат запроса

8.4.14.1. Упорядочивание по иерархии

Для справочников можно назначать упорядочивание по иерархии справочника.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Справочник.Номенклатура.Наименование,
Справочник.Номенклатура.ПолноеНаименование

УПОРЯДОЧИТЬ ПО

Справочник.Номенклатура.Наименование Иерархия

Результат:

Наименование	ПолноеНаименование
⊞ Клавиатуры	
⊞ Копировальные аппараты	
... Копировальный аппарат Omega	Копировальный аппарат Omega
⊞ Мониторы	
... Монитор 15' LG Studioworks 575N	Монитор 15 LG Studioworks 575N <0.28, 50-160Hz, 128...
... Монитор 17' Philips 107S20	Монитор Philips 107S20 17' (0.27), Flat SM, HC, AGRAS, ...
... Монитор 19' Hitachi CM715ET	Монитор 19' Hitachi CM715ET FST 0.21, 1600x1200@7...
... Монитор LCD 22' M8537ZM/A	Монитор LCD 22' M8537ZM/A Apple Cinema HD Display ...
⊞ Мыши	

Рис. 347. Результат запроса (фрагмент)

Иерархически сортировать можно только по полю, но не по некоторой операции над ним: поле упорядочивания должно содержать ссылку на поле исходной таблицы - источника данных запроса.

ВНИМАНИЕ! Упорядочивание по иерархии имеет смысл задавать в том случае, если в качестве источника определена именно таблица справочника, а не какая-либо другая таблица, содержащая только ссылку на справочник.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

РасходнаяНакладная.Контрагент.Наименование,
РасходнаяНакладная.Номер,
РасходнаяНакладная.Склад

ИЗ

Документ.РасходнаяНакладная КАК РасходнаяНакладная

В данном примере иерархического упорядочивания не получится, поскольку нет связи со справочником, и группы из этого справочника в результат запроса не попадут.

Для упорядочивания по иерархии необходимо организовать соединение со справочником.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Контрагенты.Наименование КАК Наименование,
РасходнаяНакладная.Номер,
РасходнаяНакладная.Склад

ИЗ

Справочник.Контрагенты КАК Контрагенты

ЛЕВОЕ СОЕДИНЕНИЕ Документ.РасходнаяНакладная КАК РасходнаяНакладная
 ПО Контрагенты.Ссылка = РасходнаяНакладная.Контрагент
 УПОРЯДОЧИТЬ ПО
 Наименование ИЕРАРХИЯ

Результат:

Наименование	Номер	Склад
<input type="checkbox"/> Покупатели		
Автохозяйство №34	0000012	Склад отдела продаж
Алекс-2002	0000003	Витрина в офисе
Алекс-2002	0000016	Основной склад
Алекс-2002	0000008	Основной склад
Алекс-2002	0000013	Склад отдела продаж
Завод РТИ	0000005	Витрина в офисе
Завод РТИ	0000018	Склад отдела продаж
Магазин на ул. Алексеева	0000019	Склад отдела продаж
Магазин на ул. Алексеева	0000011	Склад отдела продаж
Магазин на ул. Алексеева	0000009	Основной склад
Русская одежда	0000015	Основной склад
Центр детского творчества	0000007	Основной склад
<input type="checkbox"/> Частные лица		
Большаков Андрей	0000004	Основной склад
Никитин Юрий	0000010	Склад отдела продаж

Рис. 348. Результат запроса (фрагмент)

8.4.14.2. Упорядочивание во вложенных таблицах

В предложении **УПОРЯДОЧИТЬ ПО** можно определять также и условия упорядочивания записей из вложенных таблиц; причем их можно комбинировать с условиями упорядочивания по таблице верхнего уровня.

При этом важен порядок указания полей таблицы одного уровня (вложенной или верхнего уровня) относительно друг друга, но не важно, в каком порядке указаны поля таблицы одного уровня относительно полей таблицы другого уровня: упорядочивание выполняется всегда сначала по таблице верхнего уровня, а потом по вложенной таблице.

Пример:

[Копировать в буфер обмена](#)

```
// В отчет необходимо вывести спецификацию товарных накладных
// показать сам документ, номенклатуру и количество.
// Документы требуется упорядочить по номеру,
// а состав – по наименованию товара.
```

Выбрать

```
Документ.РасходнаяНакладная.Ссылка,
Документ.РасходнаяНакладная.Состав. (Номенклатура Как Товар, Количество)
```

Упорядочить По

```
Документ.РасходнаяНакладная.Номер,
Документ.РасходнаяНакладная.Состав.Номенклатура.Наименование
```

8.4.15. Автоупорядочивание результатов

Предложение **АВТОУПОРЯДОЧИВАНИЕ** позволяет включить режим автоматического формирования полей для упорядочивания результата запроса.

Автоупорядочивание работает по следующим принципам:

- Если в запросе было указано предложение **УПОРЯДОЧИТЬ ПО**, то каждая ссылка на таблицу, находящаяся в этом предложении, будет заменена полями, по которым по умолчанию сортируется таблица (для справочников это код или наименование, для документов - дата документа). Если поле для упорядочивания ссылается на иерархический справочник, то будет применена иерархическая сортировка по этому справочнику.

- Если в запросе отсутствует предложение **УПОРЯДОЧИТЬ ПО**, но есть предложение **ИТОГИ**, тогда результат запроса будет упорядочен по полям, присутствующим в предложении **ИТОГИ** после ключевого слова **ПО**, в той же последовательности; и если итоги рассчитывались по полям-ссылкам, то по полям сортировки по умолчанию таблиц, на которые были ссылки.
- Если в запросе отсутствуют предложения **УПОРЯДОЧИТЬ ПО** и **ИТОГИ**, но есть предложение **Сгруппировать По**, тогда результат запроса будет упорядочен по полям, присутствующим в предложении, в той же последовательности; и если группировка велась по полям-ссылкам, то по полям сортировки по умолчанию таблиц, на которые были ссылки.
- Если в запросе отсутствуют предложения **УПОРЯДОЧИТЬ ПО**, **ИТОГИ** и **Сгруппировать По**, результат будет упорядочен по полям сортировки по умолчанию для таблиц, из которых выбираются данные, в порядке их появления в запросе.

8.4.16. Расчет итогов запроса

Предложение **ИТОГИ** позволяет определить, расчет каких итогов необходим в запросе. При расчете итогов вычисляются значения агрегатных функций по выборкам с одинаковыми значениями полей - контрольных точек. Итоги добавляются в результат запроса как итоговые строки.

Порядок расчета итогов запроса описывается в соответствии со следующими правилами:

<Описание итогов>

[Копировать в буфер обмена](#)

<Итоги> [**<Описание итогов>**]

<Итоги>

[Копировать в буфер обмена](#)

ИТОГИ [**<Список итоговых полей>**] **ПО** [**ОБЩИЕ**] **<Список контрольных точек>**

<Список итоговых полей>

[Копировать в буфер обмена](#)

<Итоговое поле> [, **<Список итоговых полей>** [, ...]]

<Итоговое поле>

[Копировать в буфер обмена](#)

<Агрегатная функция> | **<Выражение>** [**[КАК]** **<Псевдоним поля>**]

<Список контрольных точек>

[Копировать в буфер обмена](#)

<Контрольная точка> [, **<Контрольная точка>** [, ...]]

<Контрольная точка>

[Копировать в буфер обмена](#)

<Выражение> [**[ТОЛЬКО]** **ИЕРАРХИЯ**] | [**ПЕРИОДАМИ**(**Секунда** | **Минута** | **Час** | **День** | **Неделя** | **Месяц** | **Квартал** | **Год** | **Декада** | **Полугодие** | **<Литерал типа DATE>** | **<Идентификатор параметра>** | **<Литерал типа DATE>** | **<Идентификатор параметра>**)] [**[КАК]** **Псевдоним поля**]

Описание итогов начинается с обязательного ключевого слова **ИТОГИ**.

Список агрегатных функций содержит перечень агрегатных функций (см. [здесь](#)), которые необходимо рассчитывать в итогах.

Ключевое слово **ОБЩИЕ** означает, что необходимо сформировать итоговую строку по всему результату запроса. Описание расчета общих итогов см. [здесь](#).

Помимо общих итогов можно задать расчет итогов по контрольным точкам. Для этого после обязательного ключевого слова **ПО** необходимо указать **<Список контрольных точек>**. Каждая контрольная точка содержит выражение, вычисляемое при выполнении запроса. По каждой комбинации значений этих выражений будут рассчитаны и добавлены в результат запроса итоговые строки.

Если контрольная точка является ссылкой на справочник, возможен расчет итогов по иерархии справочника (см. [здесь](#)). Для этого после такой ссылки нужно указать обязательное ключевое слово **ИЕРАРХИЯ**.

8.4.16.1. Расчет итогов во вложенных таблицах

В настоящей версии программы не поддерживается расчет итогов по вложенным таблицам.

8.4.16.2. Итоги по иерархии

Есть возможность рассчитать итоги по иерархии. Для этого после имени поля, для которого вычисляются итоги, необходимо указать ключевое слово **ИЕРАРХИЯ**. В результате будут рассчитаны итоги по контрольным точкам и итоги по иерархии для контрольных точек.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура,
    Документ.Количество КАК Количество,
    Документ.Ссылка.Номер,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
УПОРЯДОЧИТЬ ПО
    Документ.Номенклатура
ИТОГИ
    СУММА (Количество)
ПО
    Номенклатура ИЕРАРХИЯ
```

Результат:

Номенклатура	Количество	Номер	Контрагент
<input type="checkbox"/> Системные блоки и комплектующие	24		
<input checked="" type="checkbox"/> Сист. блок Hewlett-Packard Brio BA410	9		
<input type="checkbox"/> Сист. блок Hewlett-Packard Vectra VL420	7		
... Сист. блок Hewlett-Packard Vectra VL420	1	0000010	Никитин Юрий
... Сист. блок Hewlett-Packard Vectra VL420	3	0000008	Алекс-2002
... Сист. блок Hewlett-Packard Vectra VL420	1	0000007	Центр детского творчества
... Сист. блок Hewlett-Packard Vectra VL420	1	0000018	Завод РТИ
... Сист. блок Hewlett-Packard Vectra VL420	1	0000016	Алекс-2002
<input type="checkbox"/> Сист. блок IBM NetVista A22p	4		
... Сист. блок IBM NetVista A22p	2	0000012	Автохозяйство №34
... Сист. блок IBM NetVista A22p	1	0000007	Центр детского творчества
... Сист. блок IBM NetVista A22p	1	0000014	Эльбрус
<input checked="" type="checkbox"/> Сист. блок IBM NetVista M41	4		

Рис. 349. Результат запроса (фрагмент)

При необходимости можно рассчитать только итоги значений по иерархии, без расчета итогов в контрольных точках. Для этого перед ключевым словом **ИЕРАРХИЯ** нужно указать ключевое слово **ТОЛЬКО**.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура,
    Документ.Количество КАК Количество,
    Документ.Ссылка.Номер,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
УПОРЯДОЧИТЬ ПО
    Документ.Номенклатура
ИТОГИ
```

СУММА (Количество)
 ПО
 Номенклатура ТОЛЬКО ИЕРАРХИЯ

Результат:

Номенклатура	Количе...	Номер	Контрагент
Сист. блок IBM NetVista A22p	2	0000012	Автохозяйство №34
Сист. блок IBM NetVista A22p	1	0000007	Центр детского творчества
Сист. блок IBM NetVista A22p	1	0000014	Эльбрус
Сист. блок IBM NetVista M41	1	0000012	Автохозяйство №34
Сист. блок IBM NetVista M41	1	0000007	Центр детского творчества
Сист. блок IBM NetVista M41	1	0000008	Алекс-2002
Сист. блок IBM NetVista M41	1	0000016	Алекс-2002
<input checked="" type="checkbox"/> Мониторы	38		
Монитор 15" LG Studioworks 575N	1	0000014	Эльбрус
Монитор 15" LG Studioworks 575N	1	0000016	Алекс-2002
Монитор 15" LG Studioworks 575N	1	0000004	Большаков Андрей
Монитор 15" LG Studioworks 575N	3	0000018	Завод РТИ
Монитор 15" LG Studioworks 575N	5	0000011	Магазин на ул. Алексеева

Рис. 350. Результат запроса (фрагмент)

8.4.16.3. Дополнение дат

В случае если поле, по которому рассчитываем итоги, является полем типа **Дата**, возможно дополнение результатов датами в заданном периоде. Делается это при помощи ключевого слова **ПЕРИОДАМИ**, после которого в скобках указывается вид периода (**Секунда**, **Минута**, **Час**, **День**, **Неделя**, **Месяц**, **Квартал**, **Год**, **Декада**, **Полугодие**), начальная и конечная даты интересующего периода. Если начальная и конечная даты не указаны, будут использованы первая и последняя даты, участвующие в результате.

Пример:

[Копировать в буфер обмена](#)

```
// Получить количество покупок по клиентам по часам выбранного дня
ВЫБРАТЬ
    ПриходнаяНакладная.Контрагент,
    НАЧАЛОПЕРИОДА (ПриходнаяНакладная.Дата, ЧАС) КАК Период,
    КОЛИЧЕСТВО (ПриходнаяНакладная.Ссылка) КАК КоличествоПокупок
ИЗ
    Документ.ПриходнаяНакладная КАК ПриходнаяНакладная
Сгруппировать По
    ПриходнаяНакладная.Контрагент,
    НАЧАЛОПЕРИОДА (ПриходнаяНакладная.Дата, ЧАС)
ИТОГИ
    СУММА (КоличествоПокупок)
ПО
    Период ПЕРИОДАМИ (МИНУТА, ДАТАВРЕМЯ (2006, 6, 28), ДАТАВРЕМЯ (2006, 6, 28))
```

Результат:

Контрагент	Период	КоличествоПокупок
	28.06.2006 12:00:00	11
Темп плюс	28.06.2006 12:00:00	4
Максимум	28.06.2006 12:00:00	3
Новация	28.06.2006 12:00:00	2
Николаев	28.06.2006 12:00:00	2
	28.06.2006 14:00:00	2
Темп плюс	28.06.2006 14:00:00	1
Максимум	28.06.2006 14:00:00	1

Рис. 351. Результат запроса

Такое представление результата получится, только если при обходе результата по группировке **Период** использовать в качестве источника измерения все записи периода.

8.4.16.4. Расчет общих итогов

Для расчета итогов по всей таблице в предложении **ИТОГИ** следует указать слово **ОБЩИЕ**. В этом случае будут вычислены значения агрегатных функций для всех записей таблицы.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Документ.Номенклатура,
    Документ.Количество КАК Количество,
    Документ.Ссылка.Номер,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
ИТОГИ
    СУММА (Количество)
ПО
    ОБЩИЕ
```

Результат:

Номенклатура	Количество	Номер	Контрагент
	332		
1С:Бухгалтерия 8. Базовая версия	2	0000001	Эльбрус
1С:Бухгалтерия 8.	2	0000001	Эльбрус
Монитор 15" LG Studioworks 575N	5	0000002	Эльбрус
Мышь 2-кноп A4Tech PS/2	10	0000002	Эльбрус
Мышь LOGITECH M-S48 PS/2	10	0000002	Эльбрус
Мышь OK-720 Mouse A4Tech PS/2	2	0000002	Эльбрус
1С:Зарплата и Управление Персоналом 8	1	0000003	Алекс-2002
Windows XP Home Edition Russian CD	1	0000004	Большаков Андрей
Клавиатура Apple Pro Keyboards	1	0000004	Большаков Андрей

Рис. 352. Результат запроса (фрагмент)

8.4.16.5. Совместное использование итогов и группировки

Если итоги используются совместно с группировкой и для итогов не указан список агрегатных функций, он будет автоматически формироваться из агрегатных полей списка выборки. Если запрос содержит объединение, агрегатные функции будут браться из первого запроса.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура,
    Документ.Ссылка.Контрагент КАК Контрагент,
    СУММА (Документ.Количество) КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
Сгруппировать По
    Документ.Номенклатура,
    Документ.Ссылка.Контрагент
ИТОГИ ПО
    Номенклатура,
    Контрагент
```

Результат:

Номенклатура	Контрагент	Количество
Монитор 15' LG Studioworks 575N		16
Монитор 15' LG Studioworks 575N	Эльбрус	6
Монитор 15' LG Studioworks 575N	Эльбрус	6
Монитор 15' LG Studioworks 575N	Алекс-2002	1
Монитор 15' LG Studioworks 575N	Алекс-2002	1
Монитор 15' LG Studioworks 575N	Магазин на ул. Алексеева	5
Монитор 15' LG Studioworks 575N	Магазин на ул. Алексеева	5
Монитор 15' LG Studioworks 575N	Завод РТИ	3
Монитор 15' LG Studioworks 575N	Завод РТИ	3
Монитор 15' LG Studioworks 575N	Большаков Андрей	1
Монитор 15' LG Studioworks 575N	Большаков Андрей	1
Монитор 17' Philips 107S20		4
Монитор 17' Philips 107S20	Алекс-2002	2

Рис. 353. Результат запроса (фрагмент)

8.4.16.6. Псевдонимы итогов

Полям итогов - контрольным точкам, для которых считаются итоги, можно назначить псевдоним, для последующего обращения к ним из встроеного языка предприятия. Для этого после выражения - контрольной точки необходимо указать имя псевдонима аналогично тому, как это делается в списке полей выборки.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Документ.Номенклатура КАК Номенклатура,
Документ.Количество КАК Количество,
Документ.Ссылка.Номер,
Документ.Ссылка.Контрагент

ИЗ

Документ.РасходнаяНакладная.Состав КАК Документ

УПОРЯДОЧИТЬ ПО

Документ.Номенклатура

ИТОГИ

СУММА (Количество)

ПО

Номенклатура ТОЛЬКО ИЕРАРХИЯ КАК Товары

В случае, если псевдоним не указан, система сама даст имя итогу так, чтобы оно было уникально. В приведенном выше примере итог будет иметь имя **Товары**.

8.4.17. Выражения в языке запросов

8.4.17.1. Общая информация

Во многих конструкциях языка запросов могут использоваться выражения. Выражения языка запросов описываются следующим набором правил:

<Выражение>

[Копировать в буфер обмена](#)

<Разыменование поля> |
<Агрегатная функция> |
<Встроенная функция> |
<Операция выбора> |
<Приведение типа>[.<Разыменование поля>] |
<Значение> |
<Выражение> <Бинарная операция> <Выражение> |
<Унарная операция> <Выражение> |
(<Выражение>)

<Бинарная операция>

[Копировать в буфер обмена](#)

+ | - | * | /

<Унарная операция>

[Копировать в буфер обмена](#)

- | +

В простейшем случае выражение является ссылкой на поле исходной таблицы - источника данных запроса. Ссылка может задаваться с указанием таблицы, содержащей это поле, или без указания самой таблицы. Описание разыменования полей см. [здесь](#).

Выражения в списке полей выборки, в предложениях **ИМЕЮЩИЕ**, **ИТОГИ**, **УПОРЯДОЧИТЬ ПО**, могут быть агрегатными функциями (см. [здесь](#)).

Выражение может быть встроенной функцией языка запросов (см. [здесь](#)). Могут использоваться операции выбора (см. [здесь](#)) и операции приведения типа значения (см. [здесь](#)).

В выражениях могут непосредственно указываться значения логических, числовых, строковых и др. констант; также могут использоваться значения параметров запроса (см. [здесь](#)). В выражениях к значениям соответствующих типов могут применяться бинарные и унарные операции. При использовании операции деления рекомендуется делать проверку делителя на отличие от нуля.

Типом операций **СУММА()**, **СРЕДНЕЕ()**, **ВЫРАЗИТЬ()**, *****, **+**, **-**, применяемым к числовым типам будет являться число с произвольной точностью и разрядностью.

8.4.17.2. Разыменование полей

Выражения языка запросов в простейшем случае представляют собой ссылки на поля таблиц информационной базы. В общем виде ссылки описываются следующими правилами:

<Разыменование поля>

[Копировать в буфер обмена](#)

[<Таблица>.]<Имя поля>[.<Имя поля>[...]]

<Таблица>

[Копировать в буфер обмена](#)

<Имя таблицы> | <Псевдоним источника>

Разыменование поля начинается с имени таблицы, содержащей это поле. Если имя поля уникально - существует только у одной из таблиц среди указанных в списке источников, таблица может быть опущена.

Если поле имеет ссылочный тип, язык запросов позволяет обращаться к полям таблицы, на которую ссылается поле, и так далее. Имена полей указываются через точку.

Если исходной таблице в списке источников присвоен псевдоним источника, он может использоваться вместо имени таблицы в разыменовании полей этой таблицы. В противном случае указывается имя таблицы (см. [здесь](#)).

8.4.17.3. Агрегатные функции языка запросов

В языке запросов предусмотрены агрегатные функции, которые используются при группировке результатов запроса и при подсчете итогов. Агрегатные функции предназначены для обобщения значений указанного параметра. Определены следующие агрегатные функции:

<Агрегатная функция>

[Копировать в буфер обмена](#)

СУММА (<Выражение>) |
СРЕДНЕЕ (<Выражение>) |
МИНИМУМ (<Выражение>) |
МАКСИМУМ (<Выражение>) |
КОЛИЧЕСТВО ([РАЗЛИЧНЫЕ] <Выражение> | *)

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

Накладная.Номенклатура.Наименование,
СУММА (Накладная.Сумма) КАК Сумма,
СРЕДНЕЕ (Накладная.Сумма) КАК Среднее,
МАКСИМУМ (Накладная.Сумма) КАК Максимум,
МИНИМУМ (Накладная.Сумма) КАК Минимум,
КОЛИЧЕСТВО (Накладная.Сумма) КАК Колич

ИЗ

Документ.РасходнаяНакладная.Состав КАК Накладная

Сгруппировать По

Накладная.Номенклатура,
Накладная.Номенклатура.Наименование

ИТОГИ ПО

ОБЩИЕ

Результат:

НоменклатураНаименование	Сумма	Среднее	Максимум	Минимум	Колич
	104 086,7	873,606641178571...	16 600,00	1,10	28
Клавиатура Apple Pro Keyboards	1 690	241,428571	375,00	75,00	7
Монитор 17" Philips 107S20	840	210	210,00	210,00	4
1С:Бухгалтерия 8.	280	280	280,00	280,00	1
Сист. блок IBM NetVista M41	5 800	1 450	1 450,00	1 450,00	4
Windows XP Home Edition Russian UPG CD	255	127,5	170,00	85,00	2
Монитор 15" LG Studioworks 575N	2 400	400	750,00	155,00	6
Лазерный принтер Canon LBP-810	2 100	350	600,00	300,00	6
Сист. блок Hewlett-Packard Brio BA410	18 000	3 000	6 000,00	2 000,00	6
Мышь Ice Mouse MUS-2	39,8	5,685714	13,00	1,40	7

Рис. 354. Результат запроса (фрагмент)

Агрегатные функции могут использоваться в списке полей выборки, в предложениях **ИМЕЮЩИЕ**, **ИТОГИ**, **УПОРЯДОЧИТЬ ПО**. В списке полей выборки агрегатные функции следует использовать только для получения результата. Не рекомендуется использовать агрегатные функции в списке полей выборки для сравнения.

Корректный запрос:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

КОЛИЧЕСТВО (*),
Поставщик

ИЗ

Документ.РасходнаяНакладная КАК Накладная

Сгруппировать По

Поставщик

ИМЕЮЩИЕ

КОЛИЧЕСТВО (*) > 1

Некорректный запрос:

[Копировать в буфер обмена](#)

ВЫБРАТЬ

КОГДА КОЛИЧЕСТВО (*) > 1 ТОГДА Истина ИНАЧЕ Ложь КОНЕЦ,
Поставщик

ИЗ

Документ.РасходнаяНакладная КАК Накладная

Сгруппировать По

Поставщик

Несмотря на то, что некорректный запрос может работать успешно, такие конструкции (участие агрегатной функции не для получения результата, а для проверки условия) в разделе **ВЫБРАТЬ** имеют свою специфику на разных СУБД и могут приводить к ошибкам.

СУММА

Описание:

Функция вычисляет арифметическую сумму всех попавших в выборку значений поля.

В качестве параметра функции можно указывать только поля, содержащие числовое значение.

Если поле не может содержать числовых значений, то применение функции **СУММА** к такому полю вызовет ошибку. Данная функция может быть применена к такому полю, только если поле может содержать числовые значения (имеет составной тип данных). Но если среди значений поля в выборке встретится нечисловое значение (помимо значений **NULL**), это вызовет ошибку.

СРЕДНЕЕ

Описание:

Функция вычисляет среднее значение всех попавших в выборку значений поля.

В качестве параметра функции можно указывать только ссылки на поля, содержащие числовое значение.

Если поле не может содержать числовых значений, то применение функции **СРЕДНЕЕ** к такому полю вызовет ошибку. Данная функция может быть применена к такому полю в том случае, если поле может содержать числовые значения (имеет составной тип данных). Но если среди значений поля в выборке встретится нечисловое значение (помимо значений **NULL**), это вызовет ошибку.

МИНИМУМ

Описание:

Функция вычисляет минимальное значение из всех попавших в выборку значений поля.

В качестве параметра функции можно указывать ссылки на поля, содержащие значения любого типа.

При определении минимального значения применяются правила сравнения значений, описанные в разделе «Правила сравнения значений».

МАКСИМУМ

Описание:

Функция вычисляет максимальное значение из всех попавших в выборку значений поля.

В качестве параметра функции можно указывать выражения, содержащие значения любого типа. Не может применяться к выражениям с типом **ХранилищеЗначения**, **МоментВремени** и строкам неограниченной длины.

При определении максимального значения применяются правила сравнения значений, описанные в разделе «Правила сравнения значений».

КОЛИЧЕСТВО

Описание:

Функция подсчитывает количество значений параметра, попавших в выборку. В отличие от других агрегатных функций, функция **КОЛИЧЕСТВО** допускает три способа использования:

- Функция позволяет подсчитать количество значений указанного поля, не равных **NULL**.
- Функция позволяет подсчитать количество различных значений указанного поля, не равных **NULL**. Для этого перед спецификацией поля нужно указать ключевое слово **РАЗЛИЧНЫЕ**.
- Функция позволяет подсчитать количество строк в результате запроса. Для этого в качестве параметра функции нужно указать звездочку «*».

В качестве параметра функции можно указывать ссылки на поля, содержащие значения любого типа.

Пример:

ВЫБРАТЬ
 КОЛИЧЕСТВО (*) КАК Всего,
 КОЛИЧЕСТВО (РАЗЛИЧНЫЕ Накладная.Номенклатура) КАК Разные
 ИЗ
 Документ.РасходнаяНакладная.Состав КАК Накладная

Результат:

Всего	Разные
120	28

Рис. 355. Результат запроса

8.4.17.4. Встроенные функции языка запросов

В языке запросов определены встроенные функции, которые могут использоваться в выражениях в списке полей выборки (см. [здесь](#)) и в условии отбора в предложении **ГДЕ** (см. [здесь](#)).

Определены следующие встроенные функции:

<Встроенная функция>

[Копировать в буфер обмена](#)

ПОДСТРОКА (<Выражение>, <Значение>, <Значение>) |
 ГОД (<Выражение>) |
 КВАРТАЛ (<Выражение>) |
 МЕСЯЦ (<Выражение>) |
 ДЕНЬГОДА (<Выражение>) |
 ДЕНЬ (<Выражение>) |
 НЕДЕЛЯ (<Выражение>) |
 ДЕНЬНЕДЕЛИ (<Выражение>) |
 ЧАС (<Выражение>) |
 МИНУТА (<Выражение>) |
 СЕКУНДА (<Выражение>) |
 НАЧАЛОПЕРИОДА (<Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие) |
 КОНЕЦПЕРИОДА (<Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие) |
 ДОБАВИТЬКДАТЕ (<Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие, <Выражение>) |
 РАЗНОСТЬДАТ (<Выражение>, <Выражение>, Секунда | Минута | Час | День | Месяц | Квартал | Год) |
 ТИПЗНАЧЕНИЯ (<Выражение>) |
 ПРЕДСТАВЛЕНИЕ (<Выражение>) |
 ЕСТЬNULL (<Выражение>, <Выражение>) |
 АВТОНОМЕРЗАПИСИ ()

ПОДСТРОКА

Описание:

Данная функция предназначена для выделения подстроки из строки. В функцию передаются три параметра:

- Строка, из которой необходимо выделить подстроку. Выражение, имеющее тип **Строка**.
- Позиция символа, с которого начинается выделяемая из строки подстрока. Значение типа **Число**.
- Длина выделяемой подстроки. Значение типа **Число**.

Если в качестве первого параметра фигурирует строка, то результатом функции будет строка (возможно, нулевой длины). Если в качестве первого параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

Пример:

ВЫБРАТЬ ПЕРВЫЕ 8
 Контрагенты.Наименование,
 ПОДСТРОКА (Контрагенты.Наименование, 3, 5) КАК Подстрока
 ИЗ
 Справочник.Контрагенты КАК Контрагенты

Результат:

Наименование	Подстрока
Темп плюс	мп пл
Поставщики	ставщ
Максимус	ксиму
Новация	вация
Николаев	колае
Приборпоставка	иборп
Покупатели	купат
Эльбрус	ьбрус

Рис. 356. Результат запроса

ГОД

Описание:

Данная функция предназначена для вычисления номера года из значения типа Дата.

Параметр функции - это выражение, имеющее тип Дата.

Если в качестве параметра фигурирует значение типа Дата, то результатом функции будет значение типа Число. Если в качестве параметра будет использовано значение NULL, то результатом функции также будет значение NULL. Другие значения считаются недопустимыми и вызывают состояние ошибки.

КВАРТАЛ

Описание:

Данная функция предназначена для вычисления номера квартала из значения типа Дата. Номер квартала находится в диапазоне 1–4.

Параметр функции - это выражение, имеющее тип Дата.

Если в качестве параметра фигурирует значение типа Дата, то результатом функции будет значение типа Число. Если в качестве параметра будет использовано значение NULL, то результатом функции также будет значение NULL. Другие значения считаются недопустимыми и вызывают состояние ошибки.

МЕСЯЦ

Описание:

Данная функция предназначена для вычисления номера месяца из значения типа Дата. Номер месяца находится в диапазоне 1–12.

Параметр функции - это выражение, имеющее тип Дата.

Если в качестве параметра фигурирует значение типа Дата, то результатом функции будет значение типа Число. Если в качестве параметра будет использовано значение NULL, то результатом функции также будет значение NULL. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ДЕНЬГОДА

Описание:

Данная функция предназначена для вычисления дня года из значения типа **Дата**. День года находится в диапазоне 1–366.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ДЕНЬ

Описание:

Данная функция предназначена для вычисления дня месяца из значения типа **Дата**. День месяца находится в диапазоне 1–31.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

НЕДЕЛЯ

Описание:

Данная функция предназначена для вычисления номера недели года из значения типа **Дата**.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки. При вычислении учитывается значение свойства **Первый день недели** региональных установок информационной базы (подробнее см. [здесь](#)).

ДЕНЬНЕДЕЛИ

Описание:

Данная функция предназначена для вычисления дня недели из значения типа **Дата**. День недели находится в диапазоне 1 (понедельник) - 7 (воскресенье).

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ЧАС

Описание:

Данная функция предназначена для вычисления часа суток из значения типа **Дата**. Час суток находится в диапазоне 0–23.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

МИНУТА

Описание:

Данная функция предназначена для вычисления минуты часа из значения типа **Дата**. Минута часа находится в диапазоне 0–59.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

СЕКUNДА

Описание:

Данная функция предназначена для вычисления секунды минуты из значения типа **Дата**. Секунда минуты находится в диапазоне 0–59.

Параметр функции - это выражение, имеющее тип **Дата**.

Если в качестве параметра фигурирует значение типа **Дата**, то результатом функции будет значение типа **Число**. Если в качестве параметра будет использовано значение **NULL**, то результатом функции также будет значение **NULL**. Другие значения считаются недопустимыми и вызывают состояние ошибки.

НАЧАЛОПЕРИОДА

Описание:

Функция предназначена для выделения определенной даты из заданной даты.

Параметры функции - это выражение, имеющее тип **Дата** и тип периода - одно из значений: **Минута**, **Час**, **День**, **Неделя**, **Месяц**, **Квартал**, **Год**, **Декада**, **Полугодие**. Если тип периода указан **Неделя**, то учитывается значение свойства **Первый день недели** региональных установок информационной базы (подробнее см. [здесь](#)).

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ НАЧАЛОПЕРИОДА (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), МЕСЯЦ)
```

Результат:

Поле1
01.10.2002

Рис. 357. Результат запроса

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ НАЧАЛОПЕРИОДА (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), ДЕНЬ)
```

Результат:

Поле1
12.10.2002

Рис. 358. Результат запроса

КОНЕЦПЕРИОДА

Описание:

Функция предназначена для выделения определенной даты из заданной даты.

Параметры функции - это выражение, имеющее тип **Дата** и тип периода - одно из значений: **Минута**, **Час**, **День**, **Неделя**, **Месяц**, **Квартал**, **Год**, **Декада**, **Полугодие**. Если тип периода указан **Неделя**, то учитывается значение свойства **Первый день недели** региональных установок информационной базы (подробнее см. [здесь](#)).

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ КОНЕЦПЕРИОДА (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), МЕСЯЦ)

Результат:

Поле1
31.10.2002 23:59:59

Рис. 359. Результат запроса

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ КОНЕЦПЕРИОДА (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), ГОД)

Результат:

Поле1
31.12.2002 23:59:59

Рис. 360. Результат запроса

ДОБАВИТЬКДАТЕ

Описание:

Функция предназначена для прибавления к дате некоторой величины.

Первый параметр - исходная дата, к значению которой требуется добавить заданную величину, определяемую вторым и третьим параметрами; выражение, имеющее тип **Дата**.

Второй параметр - тип увеличения, одно из значений: **Секунда**, **Минута**, **Час**, **День**, **Неделя**, **Месяц**, **Квартал**, **Год**, **Декада**, **Полугодие**. Следует учитывать, что под термином **Декада** понимается строго 10 дней, вне зависимости от даты, указанной в первом параметре функции.

Третий параметр - величина, на которую требуется увеличить дату, задаваемую первым параметром; тип **Число** (дробная часть игнорируется).

Рекомендуется третий параметр (размерность увеличения) указывать так, чтобы он соответствовал решаемой прикладной задаче. Так, если прикладная задача оперирует днями или месяцами, то следует добавлять к дате именно дни и месяцы. Использование больших чисел в качестве значений второго и третьего параметров может приводить к следующим проблемам:

- Различное выполнение перехода на зимнее/летнее время на разных СУБД.
- Выход за границу диапазона дат, поддерживаемых СУБД.
- Выход за пределы допустимых значений параметра.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ ДОБАВИТЬКДАТЕ (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), МЕСЯЦ, 1)

Результат:

Поле1
12.11.2002 10:15:34

Рис. 361. Результат запроса

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ ДобавитьКДате (ДатаВремя (2002, 10, 12, 10, 15, 34), День, 5)

Результат:

Поле1
17.10.2002 10:15:34

Рис. 362. Результат запроса

РАЗНОСТЬДАТ

Описание:

Функция предназначена для получения разницы между двумя датами.

Первый параметр - выражение типа **Дата**. Второй параметр - выражение типа **Дата**. Третий параметр - тип разности, одно из значений: **Секунда**, **Минута**, **Час**, **День**, **Месяц**, **Квартал**, **Год**.

Рекомендуется третий параметр (тип разности) указывать так, чтобы он соответствовал решаемой прикладной задаче. Так, если прикладная задача оперирует днями или месяцами, то следует вычислять разность дат именно в днях или месяцах. Не рекомендуется получать неадекватную (с точки зрения прикладной задачи) разность дат, а потом средствами языка запросов или встроенного языка приводить ее к желаемой разнице. Данная рекомендация обусловлена тем, что расчет разницы с очень высокой дискретностью (секунды, минуты, часы) может по-разному учитывать переходы на зимнее/летнее время на разных СУБД.

Не поддерживается использование данной функции при задании значения параметра виртуальной таблицы **Период**.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ РАЗНОСТЬДАТ (ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), ДАТАВРЕМЯ (2002, 10, 14, 9, 18, 06), ДЕНЬ)

Результат:

Поле1
2

Рис. 363. Результат запроса

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ РАЗНОСТЬДАТ (ДатаВремя (2002, 10, 12), ДатаВремя (2002, 11, 03), МЕСЯЦ)

Результат:

Поле1
1

Рис. 364. Результат запроса

ВНИМАНИЕ! Функция рассчитывает календарную разницу между двумя датами, поэтому ее нельзя использовать там, где необходимо рассчитать количество банковских или рабочих дней между двумя датами.

Функция ТИПЗНАЧЕНИЯ

Описание:

Функция определения типа значения в запросе.

Параметры:

Параметр функции - выражение любого типа.

Возвращаемое значение - значение типа **Тип**.

Если параметром функции является значение **Неопределено**, то возвращаемое значение также будет **Неопределено**.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ  
    ТИПЗНАЧЕНИЯ (УчетНоменклатуры.Регистратор) КАК Документ  
ИЗ  
    РегистрНакопления.УчетНоменклатуры КАК УчетНоменклатуры
```

Функция ПРЕДСТАВЛЕНИЕ

Описание:

Данная функция предназначена для получения строкового представления значения произвольного типа.

Параметр функции - выражение любого типа.

Возвращаемое значение - представление значения, тип **Строка**.

Результат работы функции не может быть использован внутри других функций, за исключением функции **ПРЕДСТАВЛЕНИЕ**.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ  
    ПРЕДСТАВЛЕНИЕ (Документ.Контрагент) КАК Получатель,  
    ПРЕДСТАВЛЕНИЕ (Документ.Дата) КАК Дата  
ИЗ  
    Документ.РасходнаяНакладная КАК Документ
```

Результат:

Получатель	Дата
Эльбрус	28.06.2006 14:19:00
Эльбрус	28.06.2006 14:30:32
Алекс-2002	28.06.2006 14:30:49
Большаков Андрей	28.06.2006 14:32:06
Завод РТИ	28.06.2006 14:32:32
Филипенко	28.06.2006 14:32:47
Центр детского творчества	28.06.2006 14:34:04
Алекс-2002	28.06.2006 14:35:37
Магазин на ул. Алексеева	28.06.2006 14:36:05
Никитин Юлий	28.06.2006 14:36:36

Рис. 365. Результат запроса

Функция ЕСТЬNULL

Описание:

Функция предназначена для замены значения **NULL** на другое значение.

Параметры функции:

- первый параметр - выражение любого типа;
- второй параметр - выражение любого типа.

Возвращаемое значение: значение первого параметра, если первый параметр не содержит значение **NULL**; значение второго параметра в противном случае.

Второй параметр будет преобразован к типу первого в том случае, если тип первого параметра является строкой или числом.

Пример:

[Копировать в буфер обмена](#)

```
// Получить сумму по полю Количество. В случае, если нет
// записей, получить 0
ВЫБРАТЬ
    ЕСТЬNULL (СУММА (РасходнаяНакладнаяСостав.Количество), 0)
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
```

АВТОНОМЕРЗАПИСИ

Описание:

Функция предназначена для формирования поля с уникальным, последовательно возрастающим значением во временной таблице. Функцию можно использовать в том случае, если необходимо создать ключевое поле во временной таблице. Значение, сформированное этой функцией, будет уникально в пределах временной таблицы, при формировании которой использовалась функция. Начальное значение счетчика зависит от используемой СУБД и, в общем случае, может быть любым. Не гарантируется, что начальное значение счетчика будет равно 1 для любой временной таблицы.

Не поддерживается использование функции **АВТОНОМЕРЗАПИСИ()** в следующих случаях:

- в запросах, содержащих **ОБЪЕДИНИТЬ** на верхнем уровне,
- в запросах, которые не формируют временную таблицу,
- вне списка выборки,
- в выражениях языка запросов.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    ОплатаПоставщику.Поставщик КАК Поставщик,
    ОплатаПоставщику.Валюта КАК Валюта,
    СУММА (ОплатаПоставщику.Сумма) КАК Сумма,
    АВТОНОМЕРЗАПИСИ() КАК Ключ
ПОМЕСТИТЬ Оплаты
ИЗ
    Документ.ОплатаПоставщику КАК ОплатаПоставщику
СГРУППИРОВАТЬ ПО
    ОплатаПоставщику.Поставщик,
    ОплатаПоставщику.Валюта
```

Результат:

В результате будет создана временная таблица **Оплаты**, где в качестве значения поля **Ключ** будет последовательно возрастающее числовое значение.

8.4.17.5. Операции выбора в языке запросов

В выражениях языка запросов могут применяться операции выбора, которые позволяют получить одно из возможных значений в соответствии с указанными условиями.

Операция выбора описывается следующим набором правил:

<Операция выбора>

[Копировать в буфер обмена](#)

```
ВЫБОР
<Альтернативы выбора>
[ИНАЧЕ <Выражение>]
КОНЕЦ
```

<Альтернативы выбора>

[Копировать в буфер обмена](#)

```
<Одиночный выбор>
[<Альтернативы выбора>]
```

<Одиночный выбор>

[Копировать в буфер обмена](#)

```
КОГДА <Логическое выражение>
ТОГДА <Выражение>
```

В операции выбора может указываться неограниченное количество альтернативных одиночных выборов **КОГДА ... ТОГДА**. Они обрабатываются в запросе последовательно. Если логическое выражение имеет значение **Истина**, обработка операции выбора завершается. Результатом операции является значение выражения, указанного после слова **ТОГДА**. Описание логических выражений см. [здесь](#).

Значение выражения, указанного после слова **ИНАЧЕ**, используется в качестве результата операции выбора в том случае, если во всех ранее указанных альтернативных одиночных выборах предикат имел значение **Ложь**.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование,
    ВЫБОР
        КОГДА Номенклатура.ЭтоГруппа = Истина
            ТОГДА "Это группа"
        КОГДА Номенклатура.ЗакупочнаяЦена > 1000
            ТОГДА "1000 - "
        КОГДА Номенклатура.ЗакупочнаяЦена > 100
            ТОГДА "100 - 1000"
        КОГДА Номенклатура.ЗакупочнаяЦена > 10
            ТОГДА "10 - 100"
        КОГДА Номенклатура.ЗакупочнаяЦена > 0
            ТОГДА "0 - 10"
        ИНАЧЕ "Не задана"
    КОНЕЦ КАК Цена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
```

Результат:

Наименование	Цена
Мониторы	Это группа
Принтеры	Это группа
Мыши	Это группа
Клавиатуры	Это группа
Программное обеспечение	Это группа
Услуги	Это группа
Монитор 15" LG Studioworks 575N	100 - 1000
Монитор 17" Philips 107S20	100 - 1000
Монитор 19" Hitachi CM715ET	100 - 1000
Монитор LCD 22" M8537ZM/A	1000 -
Сист. блок Hewlett-Packard Brio BA410	1000 -
Сист. блок Hewlett-Packard Vectra VI 420	1000 -

Рис. 366. Результат запроса (фрагмент)

8.4.17.6. Приведение типа в языке запросов

Поля исходных таблиц могут иметь составной тип. Для таких полей возникает необходимость привести значения поля к какому-либо определенному типу. В языке запросов предусмотрена возможность приведения типа. Ею можно пользоваться в списке полей выборки и в условии отбора в предложении **ГДЕ**.

<Приведение типа>

[Копировать в буфер обмена](#)

ВЫРАЗИТЬ (<Выражение> КАК <Тип значения>)

<Тип значения>

[Копировать в буфер обмена](#)

Булево |
Число [(Длина[, Точность])] |
Строка [(Длина)] |
Дата |
<Имя таблицы>

<Длина>

[Копировать в буфер обмена](#)

Число

<Точность>

[Копировать в буфер обмена](#)

Число

Выражение приводится к одному из примитивных типов или к ссылочному типу данных. В последнем случае имя таблицы указывает на соответствующую таблицу информационной базы.

Если выражение содержит в составном типе требуемый тип значения, то приведение типа считается осуществимым, и для каждого значения указанного типа результатом будет это самое значение. Для значений других типов результатом приведения типа будет значение **NULL**.

Если выражение не содержит в составном типе требуемый тип значения, то выполнение данного запроса завершится ошибкой из-за принципиальной невозможности совершить приведение типов.

8.4.17.7. Константы и параметры в языке запросов

В выражениях языка запросов могут напрямую указываться значения типа **Булево**, **Число**, **Строка** или **Дата**. Также могут использоваться значения параметров запроса.

<Значение>

[Копировать в буфер обмена](#)

ИСТИНА |
ЛОЖЬ |
<Литерал типа ЧИСЛО> |
<Литерал типа СТРОКА> |
<Литерал типа ДАТА> |
<Литерал типа ТИП> |
<Имя параметра> |
НЕОПРЕДЕЛЕНО |
NULL

<Литерал типа Число>

[Копировать в буфер обмена](#)

<Целое число> [. <Целое число>]

<Литерал типа Строка>

[Копировать в буфер обмена](#)

"<Последовательность символов>"

<Литерал типа Дата>

ДАТАВРЕМЯ (<Целое число>, <Целое число>, <Целое число>[, <Целое число>, <Целое число>, <Целое число>])

Значения типа **Булево**, **Число**, **Строка** в языке запросов задаются так же, как и во встроенном языке.

Значения типа **Дата** задаются с помощью ключевого слова **ДАТАВРЕМЯ**, после которого в скобках последовательно указываются год, месяц, день, час, минута, секунда. Последние три указывать необязательно.

ВНИМАНИЕ! Максимальная дата, которую возможно задать при помощи литерала **ДАТАВРЕМЯ** - 31.12.3999 23:59:59.

В запрос могут передаваться параметры (см. описание объекта **Запрос**). Значения параметров могут использоваться в выражениях языка запросов. Для этого необходимо указать символ "&" и после него имя параметра.

<Литерал типа Тип>

ТИП (<Имя типа>)

<Имя типа> - имя примитивного типа либо имя таблицы, тип ссылки которой нужно получить. Результатом данной конструкции будет значение типа **Тип** для указанного типа.

Пример:

```
// Получение типа "Строка"
ТИП(Строка)
// Получение типа - ссылка на справочник "Номенклатура"
```

ТИП(Справочник.Номенклатура)

Значения типа **Тип** в языке запросов можно использовать в операциях сравнения, упорядочивания, группировки.

Пример:

```
ВЫБРАТЬ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор)
ИЗ
    РегистрНакопления.УчетНоменклатуры КАК Остатки
ГДЕ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор) = ТИП(Документ.РасходнаяНакладная)
```

Возможна передача значения типа **Тип** как параметр запроса.

Пример:

```
ВЫБРАТЬ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор)
ИЗ
    РегистрНакопления.УчетНоменклатуры КАК Остатки
ГДЕ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор) = &Тип
```

При сравнении значения типа **Тип** значения упорядочиваются в следующем порядке (первый тип считается самым малым):

- **NULL**,
- **Неопределено**,
- **Булево**,
- **Число**,

- Дата,
- Строка,
- ссылка на таблицу,
- другие типы.

8.4.18. Условия в языке запросов

8.4.18.1. Общая информация

В языке запросов используются условия отбора, в соответствии с которыми осуществляется отбор данных в предложениях **ГДЕ**, **ИМЕЮЩИЕ** и **СОЕДИНЕНИЕ**. Условия описываются по следующим правилам:

<Условие отбора>

[Копировать в буфер обмена](#)

<Логическое слагаемое> [Или <Логическое слагаемое>]

<Логическое слагаемое>

[Копировать в буфер обмена](#)

<Логический сомножитель> [И <Логический сомножитель>]

<Логический сомножитель>

[Копировать в буфер обмена](#)

Не <Логический сомножитель> |
(<Условие отбора>) |
<Логическое выражение>

В простейшем случае условие является выражением, результат которого имеет значение логического типа. Логические выражения описаны в следующем разделе.

Условия могут определяться и как более сложные логические выражения, где фигурируют простые логические выражения, соединенные между собой с помощью логических операторов **И**, **Или**, **Не**.

Логические операторы имеют приоритет:

- Самый высокий приоритет имеет логический оператор **Не**.
- Следующим по приоритету является оператор **И**.
- Самый низкий приоритет у оператора **Или**.
- В условиях сначала вычисляются простые логические выражения, затем операции **Не**, затем операции **И**, в последнюю очередь - операции **Или**. Для того чтобы обеспечить другой порядок вычислений, можно использовать круглые скобки **()**.

8.4.18.2. Логические выражения в языке запросов

В языке запросов в операциях выбора и в условиях отборов используются логические выражения.

<Логическое выражение>

[Копировать в буфер обмена](#)

<Выражение> |
(<Выражение> | <Логическое выражение>) <Операция сравнения> (<Выражение> | <Логическое выражение>) |
<Выражение> [Не] В [ИЕРАРХИИ] (<Список значений>) |
<Выражение> [Не] В [ИЕРАРХИИ] (<Описание запроса>) |
<Выражение> [Не] МЕЖДУ <Выражение> И <Выражение> |
<Выражение> ЕСТЬ [Не] NULL |
<Выражение> ССЫЛКА <Имя таблицы> |
<Выражение> [Не] ПОДОБНО <Литерал типа СТРОКА>
[СПЕЦСИМВОЛ <Литерал типа СТРОКА>]

> | < | = | >= | <= | <>

<Выражение>[, <Выражение> [, ...]]

Логическим выражением может быть:

- обычное выражение языка запросов, если его результат имеет логический тип;
- операция сравнения двух выражений языка запросов; выполняется в соответствии с правилами сравнения значений, описанными в разделе «Правила сравнения значений»;
- оператор проверки совпадения/несовпадения значения выражения с одним из перечисленных или со значениями, содержащимися в результате другого запроса;
- оператор проверки вхождения значения выражения в диапазон;
- оператор проверки значения выражения на **NULL**;
- оператор проверки ссылочного значения выражения на ссылку на определенную таблицу;
- оператор проверки строкового значения наподобие шаблону.

При сравнении значений используются правила сравнения значений, описанные ниже.

8.4.18.3. Правила сравнения значений

Поскольку в языке запросов могут сравниваться значения разных типов, определены правила, по которым выполняется сравнение двух значений.

Данные правила используются:

- для сравнения значений в операторах сравнения;
- для определения максимального и минимального значений в агрегатных функциях **МИНИМУМ** и **МАКСИМУМ**;
- для упорядочивания записей результата запроса в соответствии с порядком, заданным в предложении **УПОРЯДОЧИТЬ ПО**.

Если типы значений отличаются друг от друга, то отношения между значениями определяются на основании приоритета типов:

- тип **NULL** (самый низший);
- тип **Неопределено**;
- тип **Булево**;
- тип **Число**;
- тип **Дата**;
- тип **Строка**;
- ссылочные типы.

Отношения между различными ссылочными типами определяются на основе внутренних ссылочных номеров таблиц, соответствующих тому или иному типу.

Если типы данных совпадают, то производится сравнение значений по следующим правилам:

- У типа **Булево** значение **Истина** больше значения **Ложь**.

- У типа **Число** обычные правила сравнения для чисел.
- У типа **Дата** более ранние даты меньше более поздних.
- У типа **Строка** сравнение производится в соответствии с установленными национальными особенностями базы данных и указанным порядком сортировки. Сравнение выполняется без учета регистра символов. Операция сравнения строк не учитывает концевые пробелы, в отличие от сравнения строк на встроенном языке, где концевые пробелы участвуют в операции сравнения. Например, сравнение строк «bb» и «bb » на встроенном языке вернет **Ложь** (строки не равны), и **Истина** в языке запросов (строки равны). При этом не гарантируется, что сортировка одного и того же набора данных, выполненного на разных СУБД, вернет одинаковый результат.
- Ссылочные типы и значения типа **УникальныйИдентификатор** сравниваются на основе своих значений. При этом гарантируется повторяемость результата сравнения только в рамках одной базы данных.
- Не поддерживается сравнение полей типа **УникальныйИдентификатор** с полями других типов;
- Не допускается сравнение полей неограниченной длины (строки неограниченной длины, **ХранилищеЗначения**, поле **ТипЗначения** из таблицы планов видов характеристик).

В том случае, если используется операция сравнения двух значений, в которой участвует хотя бы одно значение **NULL**, результатом операции будет **NULL**. Если результатом условного выражения (например, предложения **ГДЕ**, **ПО**, **КОГДА**) будет **NULL**, то считается, что условие не выполняется.

8.4.18.4. Оператор проверки совпадения значения

Форма оператора «В» для проверки совпадения с одним из перечисленных

Оператор **В** позволяет проверить, совпадает ли значение выражения, указанного справа от него, с одним из значений, описанных слева. Если совпадает хотя бы с одним, результатом оператора будет **Истина**, иначе - **Ложь**. Применение **Не** изменяет действие оператора на обратное. Сравнение значений производится по правилам, описанным в разделе «Правила сравнения значений».

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Родитель.Наименование В
    ("Бытовая техника", "Оргтехника")
  
```

Форма оператора «В» для проверки принадлежности по иерархии

Для справочников проверка может осуществляться и на принадлежность по иерархии. Результатом оператора **В ИЕРАРХИИ** будет **Истина**, если значение выражения слева является ссылкой на элемент справочника и входит во множество значений справа или иерархически принадлежит какой-нибудь группе, содержащейся в этом множестве.

Пример:

[Копировать в буфер обмена](#)

```

// В качестве параметра Группа в запрос передается ссылка
// на какую либо группу справочника Номенклатура.
ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Ссылка В ИЕРАРХИИ (&Группа)
  
```

В качестве множества значений, на совпадение с которыми выполняется проверка, может фигурировать и результат запроса. В этом случае справа от оператора **В** необходимо указать

описание запроса.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Ссылка В ИЕРАРХИИ
        (ВЫБРАТЬ
            Номенклатура.Ссылка
        ИЗ
            Справочник.Номенклатура КАК Номенклатура
        ГДЕ
            Номенклатура.Наименование = "Одежда")
```

Форма оператора «В» для проверки совпадения значения с одним из результатов запроса

Пример применения данного оператора:

Пример:

[Копировать в буфер обмена](#)

```
// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
ГДЕ
    Товары.Ссылка В
        (ВЫБРАТЬ
            РасходнаяНакладнаяСостав.Номенклатура
        ИЗ
            Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав)
```

Результат:

Наименование
Монитор 15" LG Studioworks 575N
Монитор 17" Philips 107S20
Монитор 19" Hitachi CM715ET
Монитор LCD 22" M8537ZM/A
Сист. блок Hewlett-Packard Brio BA410
Сист. блок Hewlett-Packard Vectra VL420
Сист. блок IBM NetVista A22p
Сист. блок IBM NetVista M41
Лазерный принтер Canon LBP-810
Лазерный принтер 5250197-203 Minolta-QMS
Лазерный принтер HP LaserJet 2200
Мышь 2-кноп A4Tech PS/2

Рис. 367. Результат запроса (фрагмент)

Для получения противоположного результата, то есть если нужно определить, что значение не совпадает ни с одним из результатов запроса, запрос выглядит следующим образом.

Пример:

[Копировать в буфер обмена](#)

```
// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
```



```

ГДЕ
    (Не Товары.Ссылка В
      (ВЫБРАТЬ
        РасходнаяНакладнаяСостав.Номенклатура
      ИЗ
        Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав))

```

Результат:

Наименование
Мониторы
Принтеры
Мыши
Клавиатуры
Программное обеспечение
Услуги
Windows
Доставка
Инсталляция ПО
Консультации по настройке ОС Windows
Консультации по настройке 1С
Ноутбуки

Рис. 368. Результат запроса (фрагмент)

Заметим, что из запроса операции **В** возможно обращение к полям таблиц, которые встречались во внешнем запросе до появления операции.

Пример:

[Копировать в буфер обмена](#)

```

// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
ГДЕ
    Товары.Ссылка В
      (ВЫБРАТЬ
        РасходнаяНакладнаяСостав.Номенклатура
      ИЗ
        Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
      ГДЕ
        РасходнаяНакладнаяСостав.Номенклатура = Товары.Ссылка)

```

Результат:

Наименование
Монитор 15" LG Studioworks 575N
Монитор 17" Philips 107S20
Монитор 19" Hitachi CM715ET
Монитор LCD 22" M8537ZM/A
Сист. блок Hewlett-Packard Brio BA410
Сист. блок Hewlett-Packard Vectra VL420
Сист. блок IBM NetVista A22p
Сист. блок IBM NetVista M41
Лазерный принтер Canon LBP-810
Лазерный принтер 5250197-203 Minolta-QMS
Лазерный принтер HP LaserJet 2200
Мышь 2-кноп A4Tech PS/2

Рис. 369. Результат запроса (фрагмент)

Результат выражения оператора **В** с подзапросом может быть двужначным (**Истина/Ложь**). Это означает, что третий вариант результата (**NULL** - результат сравнения с **NULL**) может быть трактован как ложный результат.

В случае, если у оператора **В** с подзапросом аргумент, стоящий слева от **В** и аргумент, находящийся в подзапросе, имеют разные типы, результат **NULL** исключается из возможных результатов оператора **В**. Результат **NULL** в этом случае трактуется как ложный результат.

Использование операции «В» («Не В») по нескольким полям

Синтаксис для вложенного запроса:

(выражение1, выражение2, ..., выражениеN) В (ВЫБРАТЬ выражение1, выражение2, ..., выражениеN

Синтаксис для таблицы значений:

(выражение1, выражение2, ..., выражениеN) В (&Параметр)

В качестве параметра следует передавать таблицу значений, первые N колонок которой будут использоваться для операции **В**.

8.4.18.5. Оператор проверки вхождения значения в диапазон

Оператор **МЕЖДУ** позволяет проверить, входит ли значение выражения, указанного слева от него, в диапазон, указанный справа. Если входит, результатом оператора будет **Истина**, иначе - **Ложь**. Применение **Не** изменяет действие оператора на обратное. Правила сравнения значений см. [здесь](#).

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование,
    Номенклатура.ЗакупочнаяЦена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.ЗакупочнаяЦена МЕЖДУ 100 И 1000
```

8.4.18.6. Оператор проверки значения на NULL

Оператор **ЕСТЬ NULL** позволяет проверить значение выражения слева от него на **NULL**. Если значение равно **NULL**, результатом оператора будет **Истина**, иначе - **Ложь**. Применение **Не** изменяет действие оператора на обратное.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Товары.Наименование КАК Наименование,
    Цены.Цена КАК Цена
ИЗ
    Справочник.Номенклатура КАК Товары
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЦеныНоменклатуры КАК Цены
    ПО Товары.Ссылка = Цены.Номенклатура
ГДЕ
    Цены.Цена ЕСТЬ NULL
```

8.4.18.7. Оператор проверки ссылочного значения

Оператор **ССЫЛКА** позволяет проверить, является ли значение выражения, указанного слева от него, ссылкой на таблицу, указанную справа. Если да, результатом оператора будет **Истина**, иначе - **Ложь**. Описание разыменования таблиц см. [здесь](#).

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    РасходнаяНакладная.Номер,
    РасходнаяНакладная.Дата
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
```

8.4.18.8. Оператор проверки строки на подобие шаблону

Оператор **ПОДОБНО** позволяет сравнить значение выражения, указанного слева от него, со строкой шаблона, указанной справа. Значение выражения должно иметь тип **Строка**. Если значение выражения удовлетворяет шаблону, результатом оператора будет **Истина**, иначе - **Ложь**.

Следующие символы в строке шаблона являются служебными и имеют смысл, отличный от символа строки:

- **«%»** (процент): последовательность, содержащая любое количество произвольных символов.
- **«_»** (подчеркивание): один произвольный символ.
- **«[...]»** (в квадратных скобках один или несколько символов): любой одиночный символ из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например, **a-z**, означающие произвольный символ, входящий в диапазон, включая концы диапазона.
- **«[^...]»** (в квадратных скобках значок отрицания, за которым следует один или несколько символов): любой одиночный символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки.

Если в качестве самого себя необходимо записать один из перечисленных символов, то ему должен предшествовать спецсимвол. Сам спецсимвол (любой подходящий символ) определяется в этом же операторе после ключевого слова **СПЕЦСИМВОЛ**.

Например, шаблон:

[Копировать в буфер обмена](#)

```
"%АВВ[0 9][абвг]\_абв%" СПЕЦСИМВОЛ "\"
```

означает подстроку, состоящую из последовательности символов:

- буквы А,
- буквы Б,
- буквы В,
- одной цифры,
- одной из букв а, б, в или г,
- символа подчеркивания,
- буквы а,
- буквы б,
- буквы в.

Причем перед этой последовательностью может располагаться произвольный набор символов.

При работе в клиент-серверном варианте, возможна ситуация, когда запрос, содержащий оператор **ПОДОБНО**, не может быть выполнен. Такая ситуация возможна в том случае, когда длина строки с шаблоном превышает некоторое число, зависящее от используемой СУБД.

8.5. Выполнение и работа с запросами во встроенном языке

Для формирования запросов, выборки и обработки результатов запросов в языке предусмотрен специальный набор объектов. С помощью этих объектов выполняется формирование запроса, обход записей запроса и т. д.

Тип колонки результата запроса определяется на основе типов значений выражений, входящих в колонку и всегда включает тип `NULL`.

8.5.1. Основные приемы работы

8.5.1.1. Рабочие данные

Основные приемы работы с запросами во встроенном языке системы «1С:Предприятие» удобнее всего рассматривать на примерах. Приведем типичный пример использования запроса.

[Копировать в буфер обмена](#)

```
// Создадим Запрос
Запрос = Новый Запрос ("ВЫБРАТЬ Товар.Наименование Наименование,
    |Товар.Родитель.Наименование НаименованиеРодителя
    |ИЗ Справочник.Товары Товар");
// Выполним запрос и запишем результат в переменную РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить();
// Получим выборку из результата запроса.
Выборка = РезультатЗапроса.Выбрать();
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
    // ... выведем в окно сообщений поля из результата.
    Товар = Выборка.Наименование;
    Родитель = Выборка.НаименованиеРодителя;
    Сообщить ("Товар: " + Товар + " Родитель: " + Родитель);
КонецЦикла;
```

Как видно из этого примера, работа с запросом ведется при помощи трех основных объектов:

- `Запрос` - объект, выполняющий сам запрос. Представлен в примере переменной с именем `Запрос`.
- `РезультатЗапроса` - объект, содержащий полученные при выполнении запроса данные. Представлен в примере переменной с именем `РезультатЗапроса`.
- `ВыборкаИзРезультатаЗапроса` - объект, позволяющий обходить (т. е. перебирать) записи из результата. Представлен в примере переменной с именем `Выборка`.

Рассмотрим подробнее объект `ВыборкаИзРезультатаЗапроса`.

Для этого нам понадобится следующий запрос:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
    РасходнаяНакладнаяСостав.Количество КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
УПОРЯДОЧИТЬ ПО
    РасходнаяНакладнаяСостав.Номенклатура
ИТОГИ
    СУММА (Количество)
ПО
    Номенклатура ИЕРАРХИЯ
```

Результат:

№	Номенклатура	Количество
1	Сантехника	104
2	<i>Кран</i>	<i>84</i>
3	Кран	22
4	Кран	44
5	Кран	10
6	Кран	8
7	<i>Смеситель</i>	<i>20</i>
8	Смеситель	5
9	Смеситель	1
10	Смеситель	14
11	Мебель	134
12	<i>Стол</i>	<i>26</i>
13	Стол	1
14	Стол	15
15	Стол	10
16	<i>Стул</i>	<i>108</i>
17	Стул	55
18	Стул	5
19	Стул	32
20	Стул	16

Рис. 370. Рабочая выборка

В этой таблице мы добавили первый столбец, которого нет в тексте запроса, но он будет использоваться нами в дальнейшем для идентификации записи в результате. Итоговые записи в таблице выделены курсивом, итоговые записи для иерархических уровней справочника выделены жирным шрифтом.

8.5.1.2. Способы обхода результата запроса

8.5.1.2.1. Линейный обход результата

Первый и самый простой способ обхода - линейный. При линейном обходе выборка будет выдавать записи в той последовательности, в которой они располагаются в результате запроса. В нашем примере это будут записи с номерами 1, 2, 3, 4, 5 и так далее до записи с номером 20.

Для получения линейной выборки из результата необходимо вызвать метод `Выбрать()` объекта `РезультатЗапроса` без параметров либо с параметром `ОбходРезультатаЗапроса.Прямой`.

Пример:

[Копировать в буфер обмена](#)

```
СпособВыборки = ОбходРезультатаЗапроса.Прямой;
Выборка1 = РезультатЗапроса.Выбрать (СпособВыборки) ;
// что равнозначно записи
Выборка1 = РезультатЗапроса.Выбрать () ;
```

8.5.1.2.2. Иерархический обход результата

Следующий способ обхода результата - иерархический. При данном способе обходятся только записи, находящиеся на одном уровне. Для получения иерархической выборки из результата необходимо вызвать метод `Выбрать()` объекта `РезультатЗапроса` с параметром `ОбходРезультатаЗапроса.ПоГруппировкамСИерархией`.

Пример:

[Копировать в буфер обмена](#)

```
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
Выборка2 = РезультатЗапроса.Выбрать (СпособВыборки) ;
```

Выборка из результата с иерархическим обходом в нашем примере обойдет только записи с номерами 1 и 11, так как только эти две записи находятся на самом верхнем уровне.

Проиллюстрируем это, представив наш результат в виде дерева, где узлами будут итоговые записи, а листьями дерева - детальные записи. Вот что у нас получится:

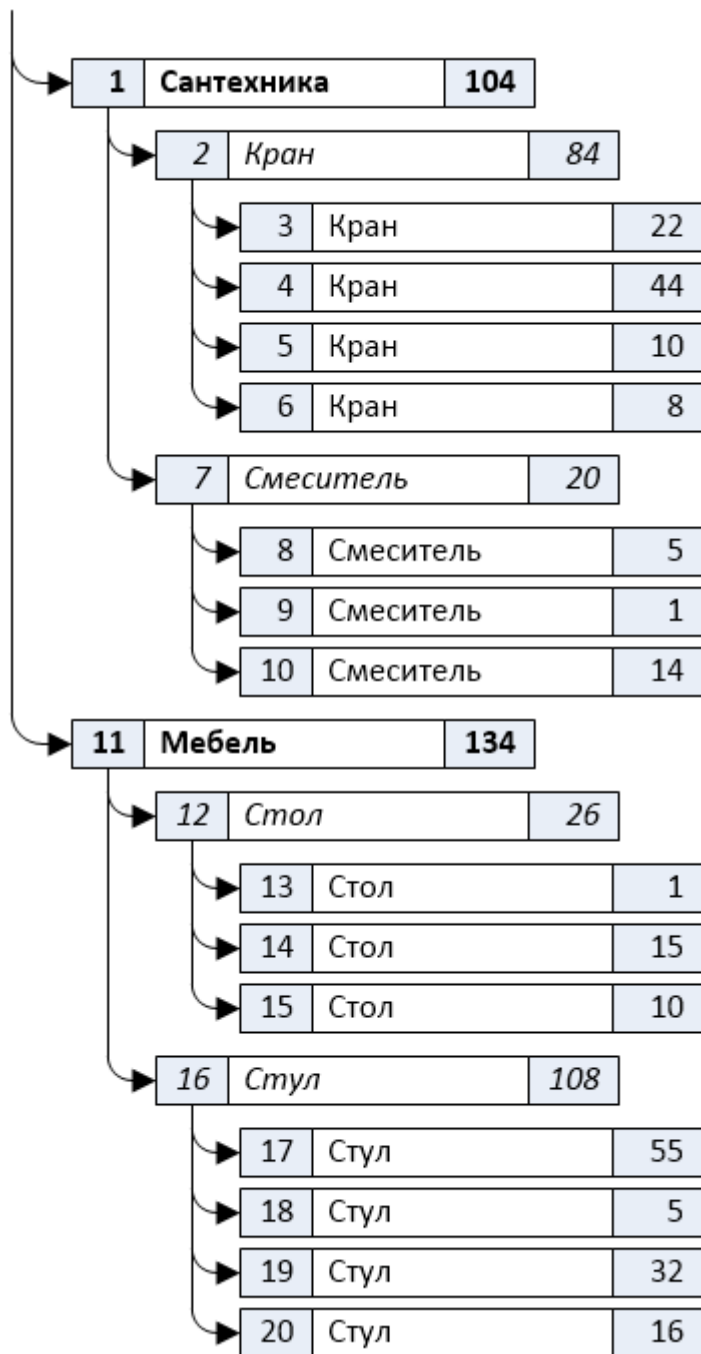


Рис. 371. Иерархический обход результата запроса

Из этого рисунка видно, что именно записи с номерами 1 и 11 и только они находятся на первом уровне дерева, в результате чего только они и попадают в первый проход иерархической выборки.

Возникает вопрос: как получать остальные записи результата запроса? Для этого у объекта `ВыборкаИзРезультатаЗапроса` можно получить еще одну выборку, которая будет обходить подчиненные записи текущей записи выборки. В нашем примере в момент, когда объект `Выборка2` будет позиционирован на запись с номером 1, мы запросим у него иерархическую выборку. Таким образом мы получим выборку, которая нам вернет записи с номерами 2, 7. А когда `Выборка2` будет позиционирована на записи с номером 11, полученная у нее иерархическая выборка вернет записи с номерами 12, 16. Так реализуется иерархический обход результатов запроса.

Заметим, что у выборки можно получать вложенные выборки любого типа. Так, если бы мы запросили у объекта `Выборка2`, позиционированной на записи 1, линейную выборку, то с ее помощью мы бы получили записи с номерами со 2-го по 10-й. Проиллюстрируем описанную методику на примере.

Пример:

[Копировать в буфер обмена](#)

```
Процедура ВыполнитьЗапрос ()
// Создадим запрос.
Запрос = Новый Запрос;
// Установим текст запроса
Запрос.Текст = "ВЫБРАТЬ
|РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
|РасходнаяНакладнаяСостав.Количество КАК Количество
|ИЗ
|Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
|УПОРЯДОЧИТЬ ПО
|РасходнаяНакладнаяСостав.Номенклатура
|ИТОГИ
|СУММА (Количество)
|ПО
|Номенклатура ИЕРАРХИЯ";
// Выполним запрос и запишем результат в переменную
// РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить ();
// Получим выборку из результата запроса.
Выборка =
РезультатЗапроса.Выбрать (ОбходРезультатаЗапроса.ПоГруппировкамСИерархией);
ВыдатьРекурсивно (Выборка);
КонецПроцедуры
Процедура ВыдатьРекурсивно (Выборка)
// Пока в выборке есть записи ...
Пока Выборка.Следующий () Цикл
// ... выведем в окно сообщений поля из результата
Товар = Выборка.Номенклатура;
Количество = Выборка.Количество;
Сообщить ("Товар: " + Товар + " Количество: " + Количество);
// Продолжим выборку подчиненных записей
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
ВыдатьРекурсивно (Выборка.Выбрать (СпособВыборки, Выборка.Группировка ()) );
КонецЦикла;
КонецПроцедуры
```

8.5.1.2.3. Обход результата по группам

Третий и последний способ обхода результата - по группам. Он сходен с иерархическим обходом, но с одним различием: записи с иерархическими итогами при обходе в нем рассматриваются как детальные, а не как итоговые. Для получения выборки по группам из результата запроса необходимо вызвать метод `Выбрать ()` объекта `РезультатЗапроса` с параметром

`ОбходРезультатаЗапроса.ПоГруппировкам`.

Пример:

[Копировать в буфер обмена](#)

```
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкам;
Выборка2 = РезультатЗапроса.Выбрать (СпособВыборки);
// Перебрав в ней все, мы получим записи с номерами:
// 1, 2, 7, 11, 12, 16.
```

Пример:

[Копировать в буфер обмена](#)

```
Процедура ВыполнитьЗапрос ()
// Создадим запрос.
Запрос = Новый Запрос;
// Установим текст запроса
Запрос.Текст = "ВЫБРАТЬ
|РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
|РасходнаяНакладнаяСостав.Количество КАК Количество
|ИЗ
|Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
|УПОРЯДОЧИТЬ ПО
```

```

| РасходнаяНакладнаяСостав.Номенклатура
| ИТОГИ
| СУММА (Количество)
| ПО
| Номенклатура ИЕРАРХИЯ";
// Выполним запрос и запишем результат в переменную
// РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить();
// Получим выборку из результата запроса
Выборка = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
    // ... выведем в окно сообщений поля из результата
    Товар = Выборка.Номенклатура;
    Количество = Выборка.Количество;
    Сообщить("Товар: "+Товар+" Итого по товару: "+Количество);
    ВыдатьДочерниеЗаписи(Выборка.Выбрать());
КонецЦикла;
КонецПроцедуры
Процедура ВыдатьДочерниеЗаписи(Выборка)
    // Пока в выборке есть записи ...
    Пока Выборка.Следующий() Цикл
        // ... выведем в окно сообщений поля из результата
        Товар = Выборка.Номенклатура;
        Количество = Выборка.Количество;
        Сообщить("Товар: "+Товар+" Количество: "+Количество);
    КонецЦикла;
КонецПроцедуры

```

8.5.1.3. Работа с выборкой

Объект `ВыборкаИзРезультатаЗапроса` предназначен для обхода записей результата запроса. Можно представить себе выборку как некоторый объект, который содержит указатель на текущую запись в результате и предоставляет программе доступ ко всем полям текущей записи. Для навигации по записям запроса используются три метода:

- `Следующий()` - перейти к следующей записи результата в соответствии с порядком обхода выборки. При первом вызове позиционирует выборку на первую запись. Когда будут выбраны все записи, данный метод просигнализирует об этом, вернув значение `Ложь`.
- `СледующийПоЗначениюПоля()` - получить следующую запись со значением в заданном поле, отличающимся от значения в этом поле текущей записи.
- `НайтиСледующий()` - найти запись с заданными значениями некоторых полей.

8.5.1.3.1. Использование метода «СледующийПоЗначениюПоля()»

Метод позволяет сгруппировать записи результата по значениям полей.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Док.Номенклатура,
    Док.Ссылка.Контрагент КАК Контрагент,
    Док.Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Док
УПОРЯДОЧИТЬ ПО
    Док.Номенклатура.Наименование,
    Контрагент

```

Результат:

№	Номенклатура	Контрагент	Количество
1	Кран	Магазин "Гигант"	10,00
2	Кран	Магазин "Хозяин"	8,00
3	Кран	Мосгорторг	44,00
4	Кран	Магазин "Мебель"	22,00
5	Смеситель	Магазин "Гигант"	5,00
6	Смеситель	Магазин "Хозяин"	1,00
7	Смеситель	Мосгорторг	14,00
8	Стол	Магазин "Гигант"	1,00
9	Стол	Магазин "Хозяин"	15,00
10	Стол	Мосгорторг	10,00
11	Стул	Магазин "Гигант"	55,00
12	Стул	Магазин "Хозяин"	5,00
13	Стул	Мосгорторг	32,00
14	Стул	Магазин "Мебель"	16,00

Рис. 372. Результат запроса

Получим линейную выборку из результата запроса и обойдем выборку при помощи метода `СледующийПоЗначениюПоля()`.

Пример:

[Копировать в буфер обмена](#)

```
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.СледующийПоЗначениюПоля("Номенклатура") Цикл
    // здесь мы получим записи с номерами 1, 5, 9, 12
    Пока Выборка.СледующийПоЗначениюПоля("Контрагент") Цикл
        // здесь мы сначала получим записи с номерами 1, 2, 3, 4
        // затем 5, 6, 7
        // затем 9, 10, 11
        // затем 12, 13, 14, 15
    КонецЦикла;
КонецЦикла;
```

Следует обратить внимание на то, что во внутреннем цикле не была выбрана запись с номером 8, т. к. в ней такое же значение поля `Получатель`, как и в предыдущей записи.

Заметим, что если в цикле получения по значению поля получать записи при помощи метода `Следующий()`, то будут выбраны все записи с равным значением поля, заданного в предыдущем вызове метода `СледующийПоЗначениюПоля()`.

Пример:

[Копировать в буфер обмена](#)

```
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.СледующийПоЗначениюПоля("Номенклатура") Цикл
    // здесь мы получим записи с номерами 1, 5, 9, 12
    Пока Выборка.Следующий() Цикл
        // здесь мы сначала получим записи с номерами 1, 2, 3, 4
        // затем 5, 6, 7, 8
        // затем 9, 10, 11
        // затем 12, 13, 14, 15
    КонецЦикла;
КонецЦикла;
```

8.5.1.3.2. Методы определения типа текущей записи

В тот момент, когда выборка позиционирована на записи, мы можем у выборки узнать характеристики этой записи. Получение характеристик записи осуществляется следующими методами:

- `Уровень()` - определяет уровень записи в результате запроса.
- `ТипЗаписи()` - определяет принадлежность записи к одному из следующих типов:

- групповой итог,
- итого по иерархии,
- детальная запись,
- общий итог.
- Группировка () - определяет имя поля, по которому были рассчитаны итоги.

Для иллюстрации работы этих методов посмотрим, что они будут возвращать в виде записей для запроса, рассматриваемого в начале главы.

[Копировать в буфер обмена](#)

ВЫБРАТЬ

РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
РасходнаяНакладнаяСостав.Количество КАК Количество

ИЗ

Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав

УПОРЯДОЧИТЬ ПО

РасходнаяНакладнаяСостав.Номенклатура

ИТОГИ

СУММА (Количество)

ПО

Номенклатура ИЕРАРХИЯ

Результат:

№	Номенклатура	Количество	Уровень	Тип записи	Группировка
1	Сантехника	104	0	Итог по иерархии	Номенклатура
2	Кран	84	1	Итог по группировке	Номенклатура
3	Кран	22	2	Детальная запись	
4	Кран	44	2	Детальная запись	
5	Кран	10	2	Детальная запись	
6	Кран	8	2	Детальная запись	
7	Смеситель	20	1	Итог по группировке	Номенклатура
8	Смеситель	5	2	Детальная запись	
9	Смеситель	1	2	Детальная запись	
10	Смеситель	14	2	Детальная запись	
11	Мебель	134	0	Итог по иерархии	Номенклатура
12	Стол	26	1	Итог по группировке	Номенклатура
13	Стол	1	2	Детальная запись	
14	Стол	15	2	Детальная запись	
15	Стол	10	2	Детальная запись	
16	Стул	108	1	Итог по группировке	Номенклатура
17	Стул	55	2	Детальная запись	
18	Стул	5	2	Детальная запись	
19	Стул	32	2	Детальная запись	
20	Стул	16	2	Детальная запись	

Рис. 373. Иерархия записей

8.5.2. Работа с временными таблицами

8.5.2.1. Общая информация

Язык запросов системы «1С:Предприятие» позволяет использовать временные таблицы в запросах. Использование временных таблиц помогает повысить производительность запросов и сделать текст сложных запросов более легким для восприятия.

Работа с временными таблицами обеспечивается двумя составляющими:

- объектом встроенного языка МенеджерВременныхТаблиц, который хранит в себе данные временных таблиц;

- синтаксисом языка запросов, позволяющим создавать новые временные таблицы и использовать существующие временные таблицы.

8.5.2.2. Менеджер временных таблиц

Менеджер временных таблиц предназначен для управления временем существования временных таблиц, создаваемых в процессе работы прикладного решения.

В одном прикладном решении может быть создано произвольное количество экземпляров менеджера временных таблиц, каждый из которых хранит свой набор временных таблиц. Каждая временная таблица однозначно идентифицируется своим именем, и в пределах одного менеджера временных таблиц все временные таблицы должны иметь уникальные имена.

ПРИМЕЧАНИЕ. Имена временных таблиц должны соответствовать требованиям, предъявляемым к именам переменных встроеного языка (см. [здесь](#)).

Экземпляр менеджера временных таблиц может быть создан с помощью конструктора **Новый**.

Пример:

[Копировать в буфер обмена](#)

```
МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;
```

При откате транзакции из менеджера временных таблиц удаляются все временные таблицы, которые были созданы в рамках транзакции и обращение к ним становится невозможным.

Все временные таблицы, созданные в данном экземпляре менеджера, существуют до тех пор, пока существует сам экземпляр менеджера временных таблиц. При уничтожении экземпляра менеджера все временные таблицы, которые содержатся в нем, также удаляются.

Менеджер временных таблиц можно закрыть принудительно при помощи метода **Закрыть ()**. При этом будут удалены все созданные в нем таблицы. Дальнейшая работа с данным экземпляром менеджера будет невозможна.

8.5.2.3. Создание временных таблиц

Создание временных таблиц осуществляется с помощью объекта **Запрос** встроеного языка системы «1С:Предприятие».

Связь запроса с менеджером временных таблиц осуществляется с помощью свойства **МенеджерВременныхТаблиц** запроса, в котором указывается тот экземпляр менеджера, в котором должны создаваться временные таблицы.

Пример:

[Копировать в буфер обмена](#)

```
МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;  
Запрос = Новый Запрос;  
Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;
```

Временная таблица может быть создана на основе данных базы данных или на основе внешнего источника данных (например, таблицы значений).

Для того чтобы создать временную таблицу на основе данных базы данных, следует установить объекту **Запрос** менеджер временных таблиц, а затем выполнить запрос к базе данных, используя ключевое слово **ПОМЕСТИТЬ**, после которого указать имя создаваемой временной таблицы. Ключевое слово **ПОМЕСТИТЬ** располагается после списка выборки запроса.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ  
    Номенклатура.Код,  
    Номенклатура.Наименование  
ПОМЕСТИТЬ ВременнаяТаблица  
ИЗ  
    Справочник.Номенклатура КАК Номенклатура
```

Во временную таблицу будут помещены значения реквизитов **Код** и **Наименования** для всех элементов справочника **Номенклатура**. При этом, с точки зрения встроенного языка, результатом исполнения такого запроса будет объект **РезультатЗапроса**, выборка из которого будет содержать одну строку с одной колонкой **Количество**. Это количество будет указывать, сколько записей было помещено в созданную временную таблицу.

Если менеджер временных таблиц не установлен, был закрыт или в установленном менеджере временных таблиц уже существует таблица с указанным именем - будет выдана ошибка.

Для того чтобы в создаваемой временной таблице добавить колонку, значение которой будет уникальным в рамках создаваемой таблицы, следует использовать функцию языка запросов **АВТОНОМЕРЗАПИСИ()**.

При создании временных таблиц не на основании внешнего источника можно использовать конструкцию **ДЛЯ ИЗМЕНЕНИЯ**. Это необходимо в тех случаях, когда требуется получить данные во временную таблицу и одновременно заблокировать их от чтения другими транзакциями.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    РасходнаяНакладная.Ссылка,
    РасходнаяНакладная.Номер,
    РасходнаяНакладная.Дата
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
ГДЕ
    РасходнаяНакладная.Ссылка В (&Документы)
ДЛЯ ИЗМЕНЕНИЯ
```

При необходимости создания индекса для временной таблицы следует в запросе указать ключевое слово **ИНДЕКСИРОВАТЬ ПО**, после которого перечислить поля, по которым нужно построить индекс.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Код КАК Код,
    Номенклатура.Наименование
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ИНДЕКСИРОВАТЬ ПО
    Код
```

Поля, по которым происходит индексирование, должны находиться в списке выборки.

Если в качестве источника используется таблица значений, то у этой таблицы значений должны быть явно указаны типы значений, содержащихся в колонках.

Для того чтобы создать временную таблицу и заблокировать данные таблиц, на основании которых создается временная таблица, следует использовать конструкцию **ДЛЯ ИЗМЕНЕНИЯ**.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    РасходнаяНакладная.Ссылка,
    РасходнаяНакладная.Номер,
    РасходнаяНакладная.Дата
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
ГДЕ
    РасходнаяНакладная.Ссылка В (&Документы)
ДЛЯ ИЗМЕНЕНИЯ
```

Для того чтобы создать временную таблицу на основании внешнего источника, следует в тексте запроса в списке источников указать имя параметра, в который будет помещен внешний источник. Остальной синтаксис идентичен обычному созданию временной таблицы.

В качестве внешнего источника могут выступать:

- таблица значений,
- табличная часть,
- результат запроса.

Ниже приведен пример создания временной таблицы на основе внешнего источника:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Источник.Код,
    Источник.Наименование
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    &ВнешнийИсточник КАК Источник
```

В данном примере во временную таблицу `ВременнаяТаблица` будет помещено содержимое колонок `Код` и `Наименование` из внешнего источника, например, таблицы значений, переданной в качестве параметра `ВнешнийИсточник`.

ВНИМАНИЕ! Если временная таблица создается на основании внешнего источника, в запросе нельзя использовать объединения и соединения, а также поля, являющиеся реквизитами полей таблиц, на основании которых создается временная таблица.

8.5.2.4. Использование временных таблиц

Для использования существующих временных таблиц в запросе следует установить объекту `Запрос` менеджер временных таблиц, после чего к временным таблицам, содержащимся в данном менеджере временных таблиц, можно обращаться по имени, как к обычным таблицам запроса.

8.5.2.5. Удаление временных таблиц

Для удаления временной таблицы из менеджера временных таблиц следует воспользоваться ключевым словом языка запроса `УНИЧТОЖИТЬ`, после которого указывается имя уничтожаемой таблицы, например:

[Копировать в буфер обмена](#)

```
УНИЧТОЖИТЬ ВременнаяТаблица
```

Если уничтожаемой таблицы не существует, будет выдана ошибка.

8.5.2.6. Отладка запроса с временными таблицами

При отладке сложных запросов, содержащих временные таблицы, возникает необходимость анализировать их (временных таблиц) содержимое. Для этого можно воспользоваться методом `Запрос.ВыполнитьПакетСПромежуточнымиДанными()` (см. [здесь](#)) или воспользоваться менеджером временных таблиц.

ПРИМЕЧАНИЕ. Рассматриваемый запрос не имеет никакого прикладного смысла, а служит исключительно для демонстрации работы с временными таблицами в процессе отладки запроса.

Рассмотрим пример такого использования:

[Копировать в буфер обмена](#)

```
Запрос = Новый Запрос;
Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;
Запрос.Текст = "ВЫБРАТЬ
    |     ПродажиОбороты.Товар,
    |     ПродажиОбороты.КоличествоОборот
| ПОМЕСТИТЬ Продажи
```

```

| ИЗ
|   РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты
| ;
|
| //////////////////////////////////////
| ВЫБРАТЬ
|   ТоварныеЗапасыОстатки.Товар,
|   ТоварныеЗапасыОстатки.КоличествоОстаток
| ПОМЕСТИТЬ Остатки
| ИЗ
|   РегистрНакопления.ТоварныеЗапасы.Остатки КАК ТоварныеЗапасыОстатки
| ;
|
| //////////////////////////////////////
| ВЫБРАТЬ
|   Товары.Наименование,
|   Продажи.КоличествоОборот,
|   Остатки.КоличествоОстаток
| ИЗ
|   Справочник.Товары КАК Товары
|       ВНУТРЕННЕЕ СОЕДИНЕНИЕ Остатки КАК Остатки
|       ПО Товары.Ссылка = Остатки.Товар
|       ВНУТРЕННЕЕ СОЕДИНЕНИЕ Продажи КАК Продажи
|       ПО Товары.Ссылка = Продажи.Товар";
Результат = Запрос.Выполнить ();
ПродажиТЗ =
Запрос.МенеджерВременныхТаблиц.Таблицы["Продажи"].ПолучитьДанные().Выгрузить();
ОстаткиТЗ = Запрос.МенеджерВременныхТаблиц.Таблицы[1].ПолучитьДанные().Выгрузить();

```

В результате выполнения запроса свойство `Запрос.МенеджерВременныхТаблиц.Таблицы` станет содержать коллекцию временных таблиц, содержащих результаты исполнения пакетного запроса. С помощью свойства `Таблицы` можно получать доступ к каждой временной таблице, причем делать это можно как по имени, так и по индексу временной таблицы. Например, таблица значений `ПродажиТЗ` получена в результате доступа к временной таблице `Продажи` по имени (`Таблицы["Продажи"]`). В то же время формирование таблицы значений `ОстаткиТЗ` осуществляется доступом по индексу (`Таблицы[1]`).

8.5.3. Работа с пакетными запросами

Платформа системы «1С:Предприятие» позволяет работать с пакетами запросов. В пакетном запросе тексты запросов разделяются символом «;». Запросы исполняются последовательно, при этом временные таблицы, которые были созданы во время исполнения какого-либо запроса, будут существовать до окончания исполнения всего пакета запроса или до исполнения в пакете запроса, уничтожающего данную временную таблицу, например:

[Копировать в буфер обмена](#)

```

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| УчетНоменклатурыОстаткиИОбороты.Номенклатура,
| УчетНоменклатурыОстаткиИОбороты.КоличествоПриход,
| УчетНоменклатурыОстаткиИОбороты.КоличествоРасход,
| УчетНоменклатурыОстаткиИОбороты.КоличествоКонечныйОстаток
| ПОМЕСТИТЬ УчетНоменклатуры
| ИЗ
| РегистрНакопления.УчетНоменклатуры.ОстаткиИОбороты(, , Авто, , ) КАК
УчетНоменклатурыОстаткиИОбороты
| ;
|
| ВЫБРАТЬ
| УчетНоменклатуры.Номенклатура,
| УчетНоменклатуры.КоличествоРасход,
| УчетНоменклатуры.КоличествоКонечныйОстаток
| ИЗ
| УчетНоменклатуры КАК УчетНоменклатуры
| ";
Результат = Запрос.Выполнить ();

```

Первый запрос создает временную таблицу, данные из которой используются во втором запросе.

Если объекту `Запрос`, исполняющему пакетный запрос, установлен менеджер временных таблиц, временные таблицы, которые не были уничтожены в рамках пакетного запроса, сохраняются в установленном менеджере. В тексте пакетного запроса возможно использование и уничтожение временных таблиц, которые существовали в установленном менеджере временных таблиц на момент запуска пакета на исполнение.

Кроме метода `Выполнить()`, последовательно выполняющего все запросы пакета и возвращающего результат последнего запроса в пакете, платформа системы «1С:Предприятие» предоставляет еще один метод - `ВыполнитьПакет()`. Этот метод последовательно выполняет все запросы и возвращает массив результатов для каждого запроса из пакета в последовательности расположения запросов в тексте пакета. Результатом выполнения запроса на уничтожение временной таблицы является значение `Неопределено`, которое также помещается в массив результатов.

При необходимости отладки пакетного запроса, предоставляется возможность использовать метод `ВыполнитьПакетСПромежуточнымиДанными()`. В этом случае результат работы аналогичен методу `ВыполнитьПакет()` с тем отличием, что временные таблицы, сформированные в пакете запросов, будут представлены не количеством записей в соответствующей временной таблице (как при вызове метода `ВыполнитьПакет()`), а содержимым этой временной таблицы.

Таким образом, можно переработать вышеприведенный пример следующим образом:

[Копировать в буфер обмена](#)

```
Запрос = Новый Запрос;
Запрос.Текст =
    "ВЫБРАТЬ
    | УчетНоменклатурыОстаткиИОбороты.Номенклатура,
    | УчетНоменклатурыОстаткиИОбороты.КоличествоПриход,
    | УчетНоменклатурыОстаткиИОбороты.КоличествоРасход,
    | УчетНоменклатурыОстаткиИОбороты.КоличествоКонечныйОстаток
    | ПОМЕСТИТЬ УчетНоменклатуры
    | ИЗ
    | РегистрНакопления.УчетНоменклатуры.ОстаткиИОбороты(, , Авто, , ) КАК
УчетНоменклатурыОстаткиИОбороты
    | ;
    |
    | ВЫБРАТЬ
    | УчетНоменклатуры.Номенклатура,
    | УчетНоменклатуры.КоличествоРасход,
    | УчетНоменклатуры.КоличествоКонечныйОстаток
    | ИЗ
    | УчетНоменклатуры КАК УчетНоменклатуры
    | ";
РезультатПакета = Запрос.ВыполнитьПакетСПромежуточнымиДанными();
ВременнаяТаблица = РезультатПакета[0].Выгрузить();
Результат = РезультатПакета[1].Выгрузить();
```

В этом случае в переменной `ВременнаяТаблица` будет доступна таблица значений, которая содержит данные временной таблицы `УчетНоменклатуры`, а в переменной `Результат` будет доступна временная таблица с результатом работы последнего запроса пакета (фактический результат работы пакета).

8.5.4. Программная работа с запросами

Система «1С:Предприятие» предоставляет возможность формировать или модифицировать запрос из встроенного языка. Для этого существует специальный объект `СхемаЗапроса`. С помощью свойства `ПакетЗапросов` разработчик получает доступ ко всем запросам, которые в данный момент редактируются с помощью объекта `СхемаЗапроса`. Методы `УстановитьТекстЗапроса()` предназначен для заполнения объекта `СхемаЗапроса` на основании переданного текста запроса, а метод `ПолучитьТекстЗапроса()` служит для обратной операции.

В качестве примера использования объектной модели работы с запросами будет приведен программный код, необходимый для формирования следующего запроса:

[Копировать в буфер обмена](#)


```

ВЫБРАТЬ
    ТоварныеЗапасыОстатки.Товар КАК Товар,
    ТоварныеЗапасыОстатки.Склад КАК Склад,
    ТоварныеЗапасыОстатки.КоличествоОстаток КАК КоличествоОстаток,
ВЫБОР
    КОГДА ТоварныеЗапасыОстатки.КоличествоОстаток > 0
        ТОГДА "Достаточно"
    ИНАЧЕ "Недостаточно"
КОНЕЦ КАК Состояние
ИЗ
    РегистрНакопления.ТоварныеЗапасы.Остатки КАК ТоварныеЗапасыОстатки
ГДЕ
    ТоварныеЗапасыОстатки.Склад = &Склад
УПОРЯДОЧИТЬ ПО
    Товар,
    Склад,
    КоличествоОстаток
ИТОГИ
    СУММА (КоличествоОстаток)
ПО
    ОБЩИЕ,
    Товар,
    Склад

```

Код на встроенном языке:

[Копировать в буфер обмена](#)

```

Конструктор = Новый СхемаЗапроса;
Пакет1 = Конструктор.ПакетЗапросов[0];
Оператор1 = Пакет1.Операторы[0];
// добавим источник в запрос
ТаблицаРегистра =
Оператор1.Источники.Добавить ("РегистрНакопления.ТоварныеЗапасы.Остатки",
"ТоварныеЗапасыОстатки");
// добавим поля в запрос
ПолеТовар = Оператор1.ВыбираемыеПоля.Добавить ("ТоварныеЗапасыОстатки.Товар", 0);
ДоступноеПолеСклад = ТаблицаРегистра.Источник.ДоступныеПоля.Найти ("Склад");
ПолеСклад = Оператор1.ВыбираемыеПоля.Добавить (ДоступноеПолеСклад, 1);
ПолеКоличество = Оператор1.ВыбираемыеПоля.Добавить ("КоличествоОстаток", 2);
ВыражениеВыбора = Новый ВыражениеСхемыЗапроса ("ВЫБОР
    | КОГДА ТоварныеЗапасыОстатки.КоличествоОстаток > 0
    |     ТОГДА ""Достаточно""
    |     ИНАЧЕ ""Недостаточно""
    | КОНЕЦ");
ПолеВыбора = Оператор1.ВыбираемыеПоля.Добавить (Строка (ВыражениеВыбора), 3);
// зададим псевдонимы
Результат = Пакет1.Колонки.Найти (ПолеКоличество);
Результат.Псевдоним = "КоличествоОстаток";
Результат = Пакет1.Колонки[1];
Результат.Псевдоним = "Склад";
Результат = Пакет1.Колонки.Найти (ВыражениеВыбора);
Результат.Псевдоним = "Состояние";
// добавим условие
Оператор1.Отбор.Добавить (Строка (ПолеСклад) + " = &Склад");
// добавим упорядочивание
Пакет1.Порядок.Добавить (ПолеТовар);
Пакет1.Порядок.Добавить ("Склад");
Пакет1.Порядок.Добавить ("КоличествоОстаток");
// зададим контрольные точки итогов
Пакет1.ОбщиеИтоги = Истина;
Пакет1.КонтрольныеТочкиИтогов.Добавить (ПолеТовар);
Пакет1.КонтрольныеТочкиИтогов.Добавить ("Склад");
// зададим агрегатные функции для итогов
КолонкаКоличество = Пакет1.Колонки.Найти (ПолеКоличество);
Пакет1.ВыраженияИтогов.Добавить ("Сумма (" + КолонкаКоличество.Псевдоним + ")");
Текст = Конструктор.ПолучитьТекстЗапроса();

```