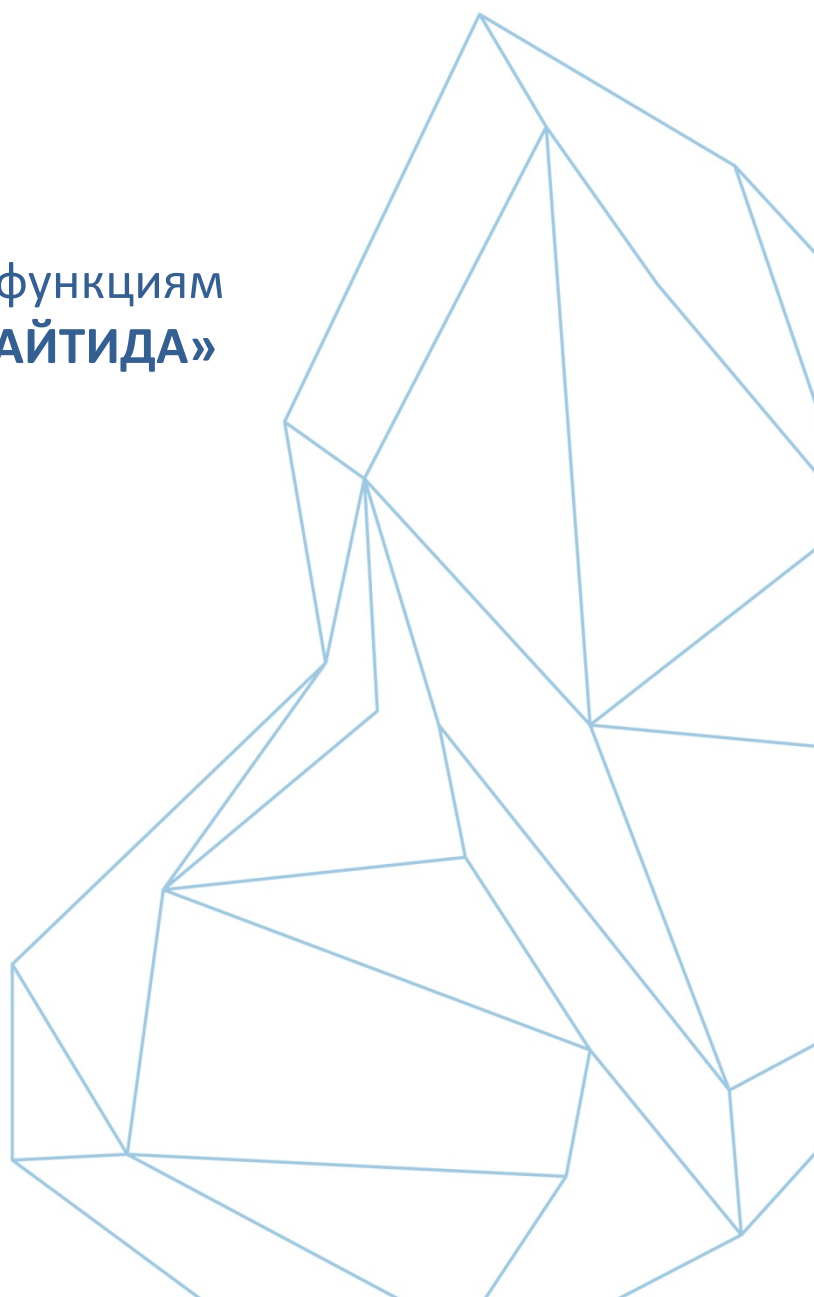




аїтїда[®]
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ДОКУМЕНТАЦИЯ
по predetermined функциям
вычислителя системы «**АЙТИДА**»



Право тиражирования принадлежит компании «Айтида».

Адрес: 192019, Санкт-Петербург, наб. Обводного канала, дом 14, корпус 4
Телефон/факс: (812) 448-8026, (812) 449-7350
Электронная почта: support@itida.ru
Адрес в сети Интернет: <http://www.itida.ru>

Автор: Вадим Соколов

Название книги: Документация по предопределенным функциям
вычислителя системы «Айтида»

Номер издания: 1.00.03

Дата выхода: 16 августа 2010 г.

Оглавление

Функции для взаимодействия с сервером и системные функции.....	8
ADDCONTEXT	8
ДОБАВИТЬКОНТЕКСТ	8
ADDMESSAGE.....	8
ДОБАВИТЬСООБЩЕНИЕ	8
APPEND	9
ДОБАВИТЬСТРОКИ.....	9
ENDOFVIEW.....	9
КОНЕЦКОНТЕКСТА	9
EVALUATE	9
ВЫЧИСЛИТЬ.....	9
GO.....	10
ПЕРЕЙТИ	10
ГОВОТТОМ.....	10
ПЕРЕЙТИВКОНЕЦ	10
ГОТОР	10
ПЕРЕЙТИВНАЧАЛО.....	10
MESSAGEBOX	11
СООБЩЕНИЕ.....	11
PROGRESSBAR	12
ИНДИКАТОР	12
QUERY	12
ЗАПРОС	12
RECCOUNT.....	13
КОЛИЧЕСТВОСТРОК.....	13
RECNO	13
НОМЕРСТРОКИ.....	13
REMOVECONTEXT	13
УДАЛИТЬКОНТЕКСТ	13
SELECTCONTEXT	13
ВЫБРАТЬКОНТЕКСТ.....	13
SENDMESSAGE	14

ПОСЛАТЬ СООБЩЕНИЕ.....	14
SKIP.....	14
ПРОПУСТИТЬ.....	14
SYSTEMMESSAGE.....	14
СИСТЕМОЕ СООБЩЕНИЕ.....	14
Примеры.....	15
Числовые функции.....	16
ABS.....	16
INT.....	16
ЦЕЛОЕ.....	16
ROUND.....	16
ОКРУГЛИТЬ.....	16
SIGN.....	17
ЗНАК.....	17
Пример.....	17
Строковые функции.....	17
ALLTRIM.....	17
СЖАТЬ ПРОБЕЛЫ.....	17
AT.....	18
ATC.....	18
LEFT.....	18
ЛЕВСИМВ.....	18
LEN.....	19
ДЛИНА.....	19
LOWER.....	19
LTRIM.....	19
СЖАТЬ ПРОБЕЛЫ ЛЕВ.....	19
PADL.....	19
ДОБАВИТЬ ПРОБЕЛЫ ЛЕВ.....	19
PADR.....	20
ДОБАВИТЬ ПРОБЕЛЫ ПРАВ.....	20
RAT.....	20
ОБРАТНЫЙ ПОИСК.....	20
REPLICATE.....	20

RIGHT.....	21
ПРАВСИМВ	21
RTRIM	21
СЖАТЬПРОБЕЛЫПРАВ	21
SPACE	21
ПРОБЕЛЫ.....	21
STR.....	22
СТРОКА	22
STRTRAN.....	22
ЗАМЕНИТЬ.....	22
STRTRANC.....	23
SUBSTR	23
ПОДСТРОКА.....	23
UPPER.....	23
VAL	24
ПРОПИСЬ	24
ПРОПИСЬРУБ.....	25
Примеры.....	25
Функции даты/времени	26
CDOW	26
ИМЯДНЯНЕДЕЛИ	26
CONSTANTSDATE	26
ДАТАКОНСТАНТ	26
STOD.....	26
DATE	26
ДАТА.....	26
DATETIME.....	27
ДАТАВРЕМЯ.....	27
DAY	27
ДЕНЬ.....	27
DOW	27
ДЕНЬНЕДЕЛИ.....	27
DYOS.....	28
GODAY	28

GOMONTH.....	29
GOYEAR	29
HOUR.....	29
MINUTE	29
МИНУТА.....	29
MONTH.....	30
МЕСЯЦ	30
MONTHFIRSTDATE	30
НАЧАЛОМЕСЯЦА.....	30
MONTHLASTDATE	30
КОНЕЦМЕСЯЦА	30
MONTHNAME	30
ИМЯМЕСЯЦА.....	30
QUARTER	31
КВАРТАЛ	31
TIME	31
ВРЕМЯ.....	31
ТТОС.....	31
ТТОD.....	32
WEEK	32
НЕДЕЛЯ	32
YEAR	32
ГОД.....	32
Примеры.....	32
Прочие функции	33
BETWEEN.....	33
МЕЖДУ	33
EMPTY	33
ПУСТО	33
IIF.....	34
ЕСЛИ.....	34
INLIST.....	34
ВСПИСКЕ	34
МАХ.....	34

MIN	35
RAND	35
СЛУЧАЙНОЕЧИСЛО	35
TYPE	35
ТИП	35
UNIQUENAME	36
УНИКАЛЬНОЕИМЯ	36
Пример	36
Управляющие конструкции	36
BREAK	36
IF...ELSE	37
RETURN	37
WHILE	37
Примеры.....	38

Функции для взаимодействия с сервером и системные функции

ADDCONTEXT(СтрокаSQL, ИмяКонтекста[, СообщениеОжидания,[НомерСоединения | СписокФилиалов]])

ДОБАВИТЬКОНТЕКСТ(СтрокаSQL, ИмяКонтекста[, СообщениеОжидания,[НомерСоединения | СписокФилиалов]])

Выполняет указанный запрос SQL и формирует таблицу контекста из результата выполнения запроса для дальнейшего использования. Функция позволяет указать номер соединения или список филиалов для выборки данных. Добавленный контекст становится текущим контекстом.

Если сервер вернул ошибку выполнения запроса, то функция возвращает Ложь и записывает полученное сообщение от сервера в журнал сообщений и ошибок (errorlog).

Параметры.

СтрокаSQL	- Текст SQL запроса, из результата выполнения которого будет сформирован контекст.
ИмяКонтекста	- Имя, которое будет присвоено полученному контексту.
СообщениеОжидания	- Текст сообщения, который будет выведен во время выполнения запроса. При продолжительном времени выполнения запросов, позволяет известить пользователя о происходящем действии. Если параметр не указан, то выводится стандартное сообщение.
НомерСоединения	- позволяет указать номер источника данных, из которого будет произведена выборка. Данный номер возвращается функцией ДобавитьСоединение.
СписокФилиалов	- Список филиалов через пробел, из баз которых будет произведена выборка данных и объединена в единый контекст.

Возвращаемое значение - логическое. Истина, если запрос был выполнен удачно и контекст был сформирован. Ложь, если сервер вернул ошибку.

ADDMESSAGE(КодСообщения[, ТекстСообщения])

ДОБАВИТЬСООБЩЕНИЕ(КодСообщения[, ТекстСообщения])

Добавляет указанное сообщение в список сообщений и ошибок отчета. Также это сообщение добавляется в журнал ошибок и сообщений системы (errorlog).

Параметры.

КодСообщения	- Строка, код сообщения. Должен начинаться с W для сообщений и с E для ошибок.
ТекстСообщения	- Строка, текст сообщения.

Возвращаемое значение - нет.

APPEND(КоличествоСтрок[, ИмяКонтекста])
ДОБАВИТЬСТРОКИ(КоличествоСтрок[, ИмяКонтекста])

Добавляет указанное количество пустых строк в указанный контекст.

Параметры.

КоличествоСтрок	- Число. Количество строк, которое необходимо добавить. За один вызов не может быть добавлено более 1000 строк.
ИмяКонтекста	- Строка. Указывает, в какой контекст необходимо добавить строки. Если параметр не указан, строки будут добавлены в текущий контекст.

Возвращаемое значение - нет.

ENDOFVIEW([ИмяКонтекста])
КОНЕЦКОНТЕКСТА([ИмяКонтекста])

Функция возвращает ИСТИНУ, если в указанном контексте указатель строк находится за пределами последней строки.

Параметры.

ИмяКонтекста	- Строка, имя контекста, для которого проводится проверка. Если параметр не указан, проверяется текущий контекст.
--------------	---

Возвращаемое значение - Логическое. Истина, если достигнут конец контекста. Иначе - Ложь.

EVALUATE(Программа)
ВЫЧИСЛИТЬ(Программа)

Выполняет набор команд, переданный в качестве аргумента. Команды передаются в виде строки текста, который анализируется и выполняется. Выполнение производится в текущем окружении, т.е. доступны все текущие переменные и контексты. Также, если в тексте Программы определяются новые переменные или изменяются значения текущих переменных, производятся какие-либо действия с контекстами, то все эти изменения отражаются на вызывающей программе. Размер программы не ограничен. Функция возвращает значение последнего вычисленного выражения.

Параметры.

Программа	- Строковое выражение, содержащее текст выполняемой программы.
-----------	--

Возвращаемое значение - зависит от выполняемой программы и является значением последнего вычисленного выражения Программы.

GO(НомерСтроки [, ИмяКонтекста])

ПЕРЕЙТИ(НомерСтроки [, ИмяКонтекста])

Выполняет переход на указанную строку в указанном контексте. Если имя контекста не указано, используется текущий контекст. Если НомерСтроки меньше или равен нулю, то переход осуществляется на первую строку контекста. Если НомерСтроки больше или равен количеству строк в контексте, то переход осуществляется на последнюю строку контекста.

Параметры.

НомерСтроки

- Число, номер строки, на которую необходимо произвести переход.

ИмяКонтекста

- Строка. Указывает, в каком контексте необходимо осуществить переход. Если параметр не указан, переход будет осуществлен в текущем контексте.

Возвращаемое значение - нет.

ГОВОТТОМ(ИмяКонтекста)

ПЕРЕЙТИВКОНЕЦ(ИмяКонтекста)

Выполняет переход на последнюю строку в указанном контексте. Если имя контекста не указано, используется текущий контекст.

Параметры.

ИмяКонтекста

- Строка. Указывает, в каком контексте необходимо осуществить переход. Если параметр не указан, переход будет осуществлен в текущем контексте.

Возвращаемое значение - нет.

ГОТОР(ИмяКонтекста)

ПЕРЕЙТИВНАЧАЛО(ИмяКонтекста)

Выполняет переход на первую строку в указанном контексте. Если имя контекста не указано, используется текущий контекст.

Параметры.

ИмяКонтекста

- Строка. Указывает, в каком контексте необходимо осуществить переход. Если параметр не указан, переход будет осуществлен в текущем контексте.

Возвращаемое значение - нет.

MESSAGEBOX([ТекстСообщения [, Заголовок[, Флаги]])
СООБЩЕНИЕ([ТекстСообщения [, Заголовок[, Флаги]])

Выводит диалоговое окно с указанным **ТекстСообщения** и **Заголовок**. **Флаги** могут содержать информацию о количестве и содержании доступных кнопок.

Параметры.

ТекстСообщения	- Строка, содержащая текст выводимого сообщения. Строка может содержать символы перевода строки (0x0D) для отображения нескольких строк текста.
Заголовок	- Строка, содержащая текст, который будет выведен в заголовок окна. Если параметр не указан, то в заголовок будет выведено слово Сообщение .
Флаги	- Число, указывающее, какой набор кнопок будет присутствовать в окне. Если параметр не указан, то будет использовано значение 0. Возможные значения параметра: 0 - одна кнопка: ОК, 1 - две кнопки: ОК и Отмена, 2 - три кнопки: Прервать, Отменить и Пропустить, 3 - три кнопки: Да, Нет и Отмена 4 - две кнопки: Да и Нет 5 - две кнопки: Повторить и Отменить.

Возвращаемое значение - Число, соответствующее нажатой пользователем кнопке.
Возможные значения:

- 1 - кнопка ОК,
- 2 - кнопка Отмена,
- 3 - кнопка Прервать,
- 4 - кнопка Повторить,
- 5 - кнопка Пропустить,
- 6 - кнопка Да,
- 7 - кнопка Нет.

POSTMESSAGE(ДескрипторОкна, Сообщение[, wParam[, lParam]])
ОТПРАВИТЬ СООБЩЕНИЕ(ДескрипторОкна, Сообщение[, wParam[, lParam]])

Отправляет указанное **Сообщение** указанному окну. Используется системная функция **PostMessage**.
В отличие от **SendMessage**, функция возвращается немедленно после отправки сообщения.

Параметры.

ДескрипторОкна	- Число целое, дескриптор окна, которому отправляется сообщение.
Сообщение	- Число целое, сообщение для отправки.
wParam	- Число целое, первый параметр сообщения.
lParam	- Число целое, второй параметр сообщения.

Возвращаемое значение - Логическое. Истина, если сообщение успешно отправлено. Иначе - Ложь.

PROGRESSBAR([Заголовок/Индикатор [, База/Подзаголовок [, Подзаголовок/Заголовок]])
ИНДИКАТОР([Заголовок/Индикатор [, База/Подзаголовок [, Подзаголовок/Заголовок]])

Выводит и управляет отображением индикатора хода выполнения работы. В зависимости от переданных параметров, функция выполняет следующие действия. Если первый параметр Заголовок имеет строковое значение, то функция инициализирует новый индикатор. В этом случае второй параметр должен указывать общее количество обрабатываемых данных, принимаемое за 100%. Третий параметр не используется. Если первый параметр числовой, то функция обновляет текущий выведенный индикатор, изменяя его отображение. При этом второй параметр указывает на подзаголовок, выводимый непосредственно над полосой индикатора и дающий словесное описание хода работы. Третий параметр может изменить основной заголовок индикатора. Если параметры не указаны, то индикатор убирается и его работа считается завершенной.

Параметры.

Заголовок	- Строка, выводимая в заголовок окна индикатора.
Индикатор	- Число, на основании которого рассчитывает процентное выполнение работы. Процент вычисляется относительно Базы, указанной при инициализации индикатора.
База	- Число, общее количество обрабатываемых данных, принимаемое за 100%.
Подзаголовок	- Строка, подзаголовок, дающий словесное описание хода работы.

Возвращаемое значение - нет.

QUERY(SQLЗапрос[, Поле[, НомерСоединения[, СписокФилиалов]])
ЗАПРОС(SQLЗапрос[, Поле[, НомерСоединения[, СписокФилиалов]])

Выполняет указанный запрос SQL и возвращает значение указанного поля первой записи результата выполнения запроса. Если имя поля не указано, то возвращается значение первого поля первой записи результата выполнения запроса. Если сервер вернул ошибку выполнения запроса, то функция возвращает Ложь и записывает полученное от сервера сообщение в журнал сообщений и ошибок (errorlog).

Параметры.

SQLЗапрос	- Строка, содержащая SQL запрос к источнику данных.
Поле	- Строка, имя поля результата выполнения запроса, значение которого необходимо вернуть. Если параметр не указан, будет возвращено значение первого поля.
НомерСоединения	- позволяет указать номер источника данных, в котором будет выполнен запрос. Данный номер возвращается функцией ДобавитьСоединение .
СписокФилиалов	- Список филиалов через пробел, в базах которых будет выполнен SQL запрос.

Возвращаемое значение - зависит от типа полученных данных.

RECOUNT([ИмяКонтекста]) **КОЛИЧЕСТВОСТРОК**([ИмяКонтекста])

Функция возвращает количество строк в указанном контексте.

Параметры.

ИмяКонтекста - Строка, имя контекста, количество строк которого определяется.
Если параметр не указан, то возвращается количество строк в текущем контексте.

Возвращаемое значение - Число целое, равное количеству строк в указанном контексте.

RECNO([ИмяКонтекста]) **НОМЕРСТРОКИ**([ИмяКонтекста])

Функция возвращает номер текущей строки в указанном контексте.

Параметры.

ИмяКонтекста - Строка, имя контекста, количество строк которого определяется.
Если параметр не указан, то возвращается количество строк в текущем контексте.

Возвращаемое значение - Число целое, равное количеству строк в указанном контексте.

REMOVECONTEXT(ИмяКонтекста) **УДАЛИТЬКОНТЕКСТ**(ИмяКонтекста)

Функция удаляет из системы контекст с указанным именем и освобождает все связанные с контекстом ресурсы. После удаления контекст становится недоступным для использования. В случае удаления контекста, по которому производится вывод секции, вывод секции прекращается на следующей итерации и производится стандартный выход из секции.

Параметры.

ИмяКонтекста - Строка, имя контекста, который необходимо удалить.

Возвращаемое значение - нет.

SELECTCONTEXT(ИмяКонтекста) **ВЫБРАТЬКОНТЕКСТ**(ИмяКонтекста)

Выбирает в качестве текущего контекста указанный контекст.

Параметры.

ИмяКонтекста - Строка. Указывает, какой контекст следует сделать текущим.

Возвращаемое значение - Логическое. Истина, если указанный контекст существует и был выбран в качестве текущего. Иначе - Ложь.

SENDMESSAGE(ДескрипторОкна, Сообщение[, wParam[, lParam]])

ПОСЛАТЬСООБЩЕНИЕ(ДескрипторОкна, Сообщение[, wParam[, lParam]])

Посылает указанное Сообщение указанному окну. Используется системная функция SendMessage.

В отличие от PostMessage, функция возвращается только после обработки сообщения указанным окном.

Параметры.

ДескрипторОкна - Число целое, дескриптор окна, которому посылается сообщение.

Сообщение - Число целое, сообщение для отправки.

wParam - Число целое, первый параметр сообщения.

lParam - Число целое, второй параметр сообщения.

Возвращаемое значение - Число целое, результат обработки сообщения функцией окна получателя сообщения.

SKIP([КоличествоСтрок [, ИмяКонтекста]])

ПРОПУСТИТЬ([КоличествоСтрок [, ИмяКонтекста]])

Выполняет переход на строку, отстоящую от текущей на указанное КоличествоСтрок в указанном контексте. Если имя контекста не указано, используется текущий контекст. Если КоличествоСтроки меньше нуля, то переход осуществляется к началу контекста, иначе к концу контекста.

Параметры.

КоличествоСтрок - Число, количество строк, которое необходимо пропустить. Если параметр не указан, то пропускается одна строка.

ИмяКонтекста - Строка. Указывает, в каком контексте необходимо осуществить переход. Если параметр не указан, переход будет осуществлен в текущем контексте.

Возвращаемое значение - нет.

SYSTEMMESSAGE([ТекстСообщения])

СИСТЕМНОЕСООБЩЕНИЕ([ТекстСообщения])

Выводит окно с текстом системного сообщения. Окно с текстом системного сообщения - это окно, которое выводится в правом верхнем углу активного окна приложения, содержащее одну строку текста, который информирует пользователя о выполняемых в данный момент действиях. Данное окно не взаимодействует с пользователем и не требует его действий.

Параметры.

ТекстСообщения - Строка, текст сообщения, которое будет выведено.

Если параметр не указан, окно с системным сообщением будет убрано с экрана.

Возвращаемое значение - нет.

Примеры

А. В данном примере функция Запрос используется для выполнения запроса, который заносит информацию о соответствии между складами, филиалами и учетами во временную таблицу.

При помощи функции ДобавитьКонтекст на основании данных во временной таблице формируется контекст с именем crsSk1, затем функция ПерейтиВНачало делает первую строку контекста текущей. Далее при помощи оператора цикла WHILE осуществляется проход по всем строкам контекста.

На каждой итерации цикла функция СистемноеСообщение в правом верхнем углу окна генерации отчета выводит сообщение со значениями полей name и dbase в текущей строке контекста, затем выполняется некоторый запрос. Переход к следующей строке контекста осуществляется с помощью функции Skip.

Условием выхода из цикла является возврат функцией КонецКонтекста значения Истина.

```
Запрос("
select a.code, a.name, a.clev, a.branch,
       cast(isnull(b.dbase, '') as varchar(30)) as dbase, isnull(c.name,
'себестоимость') as name_clev
into #temp from sprsk1 a
left outer join sprbranch b on b.code=a.branch and b.folder=0
left outer join sprccnames c on c.clev=a.clev
");
```

```
ДобавитьКонтекст("select * from #temp", "crsSk1");
ПерейтиВНачало("crsSk1");
WHILE (!КонецКонтекста("crsSk1"))
{
    СистемноеСообщение("данные по "+crsSk1.name+", БД -
"+crsSk1.dbase);
    Запрос("
--запрос, извлекающий данные из БД crsSk1.dbase
");
    skip(1, "crsSk1");
}
```

В. В данном примере функция Reccount используется для проверки того, что в контексте с именем Контекст, сформированном на основании указанных параметров отчета, имеются строки. Если это не так, пользователю выдается соответствующее сообщение.

```
IF (Reccount("Контекст")==0)
Сообщение("По заданным параметрам товара нет!");
```

Числовые функции

ABS(Число)

Возвращает абсолютное значение своего аргумента.

Параметры.

Число - Число, целое или вещественное, беззнаковое значение которого будет возвращено.

Возвращаемое значение – Число целое или вещественное, абсолютное значение указанного Числа.

INT(Число)

ЦЕЛОЕ(Число)

Возвращает целую часть числа.

Параметры.

Число - Число, целое или вещественное, целая часть которого будет возвращена.

Возвращаемое значение – Число целое, целая часть указанного Числа.

ROUND(Число, КоличествоЗнаков)

ОКРУГЛИТЬ(Число, КоличествоЗнаков)

Возвращает Число, округленное до указанного количества знаков до или после запятой. Если указанное количество знаков больше нуля, то округление производится до соответствующих долей числа. Если указанное количество знаков меньше нуля, то округление производится до соответственно десятков, сотен и т.д.

Параметры.

Число - Число, требующее округления.

КоличествоЗнаков - Число, количество знаков до или после запятой, до которого необходимо произвести округление.

Возвращаемое значение - Число вещественное.

SIGN(Число) **ЗНАК**(Число)

Функция возвращает 1, если указанное Число положительное, -1, если оно отрицательное и 0, если число равно нулю.

Параметры.

Число - Число, знак которого необходимо определить.

Возвращаемое значение - Число целое.

Пример

В данном примере функция ABS используется для получения абсолютного значения отрицательного вещественного числа. Результатом применения функции будет 12.34000000. После округления этого числа с помощью функции INT оно примет значение 12. Функция ROUND, вызванная с указанными параметрами, вернет 100.00000000, тем самым иллюстрируя, что, в отличие от ABS, она меняет тип данных своего аргумента при преобразовании. Функция SIGN применяется для получения знака числа -10. Результатом вычисления всего выражения будет 88.00000000.

Переменная = **SIGN**(-10)***INT**(**ABS**(-12.34))+**ROUND**(123,-2);
Сообщение(Переменная);

Строковые функции

ALLTRIM(Строка) **СЖАТЬПРОБЕЛЫ**(Строка)

Убирает все лидирующие и завершающие пробелы из переданной строки.

Параметры.

Строка - Строка, из которой необходимо убрать пробелы.

Возвращаемое значение - Строка, полученная из исходной, без лидирующих и завершающих пробелов.

AT(СтрокаГдеИскать, СтрокаЧтоИскать[, НомерВхождения])

Возвращает номер позиции указанного (НомерВхождения) вхождения подстроки (СтрокаЧтоИскать) в исходную строку (СтрокаГдеИскать). Если вхождений не обнаружено, будет возвращен 0. Поиск осуществляется с учетом регистра.

Параметры.

СтрокаГдеИскать	- Строка. В этой строке будет производиться поиск.
СтрокаЧтоИскать	- Подстрока, которую необходимо найти в исходной строке.
НомерВхождения	- Номер вхождения подстроки, которое необходимо найти. Если параметр не указан, будет найдено первое вхождение.

Возвращаемое значение - Число, номер позиции, начиная с 1, с которой начинается указанное вхождение подстроки в исходной строке. Если указанного вхождения не обнаружено, то функция вернет 0.

ATC(СтрокаГдеИскать, СтрокаЧтоИскать[, НомерВхождения])

Возвращает номер позиции указанного (НомерВхождения) вхождения подстроки (СтрокаЧтоИскать) в исходную строку (СтрокаГдеИскать). Если вхождений не обнаружено, будет возвращен 0. Поиск осуществляется без учета регистра.

Параметры.

СтрокаГдеИскать	- Строка. В этой строке будет производиться поиск.
СтрокаЧтоИскать	- Подстрока, которую необходимо найти в исходной строке.
НомерВхождения	- Номер вхождения подстроки, которое необходимо найти. Если параметр не указан, будет найдено первое вхождение.

Возвращаемое значение - Число, номер позиции, начиная с 1, с которой начинается указанное вхождение подстроки в исходной строке. Если указанного вхождения не обнаружено, то функция вернет 0.

LEFT(Строка, КоличествоСимволов)

ЛЕВСИМВ(Строка, КоличествоСимволов)

Возвращает указанное количество начальных символов строки. Если КоличествоСимволов меньше или равно нулю, то будет возвращена пустая строка. Если КоличествоСимволов больше или равно длине строки, будет возвращена вся строка целиком.

Параметры.

Строка	- Строка, начальные символы которой будут возвращены.
КоличествоСимволов	- Число, количество символов, которое необходимо вернуть.

Возвращаемое значение - Строка, состоящая из указанного количества начальных символов исходной строки.

LEN(Строка)
ДЛИНА(Строка)

Возвращает длину указанной строки. Если строка не содержит символов, функция возвращает 0.

Параметры.

Строка - Строка, длина которой будет возвращена.

Возвращаемое значение – Число, длина указанной строки в символах.

LOWER(Строка)

Преобразует указанную строку в строчные символы.

Параметры.

Строка - Строка, которая будет преобразована в строчные символы.

Возвращаемое значение - Строка, полученная из исходной заменой всех символов на строчные.

LTRIM(Строка)
СЖАТЬПРОБЕЛЫЛЕВ(Строка)

Убирает все лидирующие пробелы из переданной строки.

Параметры.

Строка - Строка, из которой необходимо убрать пробелы.

Возвращаемое значение - Строка, полученная из исходной, без лидирующих пробелов.

PADL(Строка, Длина)
ДОБАВИТЬПРОБЕЛЫЛЕВ(Строка, Длина)

Добавляет в начало переданной строки пробелы до достижения требуемой длины. Если длина переданной строки больше требуемой, то строка будет усечена до требуемой длины.

Параметры.

Строка - Строка, к которой будут добавлены пробелы.

Длина - Число, требуемая длина результата.

Возвращаемое значение - Строка требуемой длины.

PADR(Строка, Длина)

ДОБАВИТЬПРОБЕЛЫПРАВ(Строка, Длина)

Добавляет в конец переданной строки пробелы до достижения требуемой длины. Если длина переданной строки больше требуемой, то строка будет усечена до требуемой длины.

Параметры.

- | | |
|--------|--|
| Строка | - Строка, к которой будут добавлены пробелы. |
| Длина | - Число, требуемая длина результата. |

Возвращаемое значение - Строка требуемой длины.

RAT(Строка, Символ | Число)

ОБРАТНЫЙПОИСК(Строка, Символ | Число)

Возвращает номер позиции указанного Символа в Строке. Поиск осуществляется с конца строки. Если Символ не найдет, то возвращается 0.

Параметры.

- | | |
|--------|---|
| Строка | - Строка, в которой осуществляется поиск. |
| Символ | - Искомый символ. |
| Число | - Число с кодом искомого символа. |

Возвращаемое значение - Число целое, позиция искомого символа в строке. Если символ не найден, то возвращается 0.

REPLICATE(Символ | Строка, КоличествоПовторов)

Функция возвращает строку, состоящую из указанного количества повторений Символа либо Строки.

Параметры.

- | | |
|--------------------|---|
| Символ | - Символ, который необходимо повторить. |
| Строка | - Строка, которую необходимо повторить. |
| КоличествоПовторов | - Количество раз, которое необходимо произвести повторение. |

Возвращаемое значение - Строка, состоящая из необходимого количества повторений Символа или Строки.

RIGHT(Строка, КоличествоСимволов)

ПРАВСИМВ(Строка, КоличествоСимволов)

Возвращает указанное количество конечных символов строки. Если КоличествоСимволов меньше или равно нулю, то будет возвращена пустая строка. Если КоличествоСимволов больше или равно длине строки, будет возвращена вся строка целиком.

Параметры.

Строка - Строка, конечные символы которой будут возвращены.
КоличествоСимволов - Число, количество символов, которое необходимо вернуть.

Возвращаемое значение - Строка, состоящая из указанного количества конечных символов исходной строки.

RTRIM(Строка)

СЖАТЬПРОБЕЛЫПРАВ(Строка)

Убирает все завершающие пробелы из переданной строки.

Параметры.

Строка - Строка, из которой необходимо убрать пробелы.

Возвращаемое значение - Строка, полученная из исходной, без завершающих пробелов.

SPACE(Количество)

ПРОБЕЛЫ(Количество)

Возвращает строку, состоящую из указанного количества пробелов.

Параметры.

Количество - Число, количество требуемых пробелов.

Возвращаемое значение - Строка, состоящая из требуемого количества пробелов.

STR(Число [, Длина= 10 [,КоличествоЗнаков= 0]])

СТРОКА(Число [, Длина= 10 [,КоличествоЗнаков= 0]])

Преобразует указанное число в строку указанной длины и с указанным количеством знаков после запятой.

Параметры.

Число	- Число, которое необходимо преобразовать в строку.
Длина	- Число, требуемая длина результирующей строки. Если параметр не указан, то возвращается строка длиной 10 символов.
КоличествоЗнаков	- Число, требуемое количество знаков после запятой. Если параметр не указан, то возвращается только целая часть числа.

Возвращаемое значение - Строка, содержащая символьное представление числа.

STRTRAN(СтрокаДляПоиска, ИскомаяСтрока, Замена [, НомерФрагмента [, КоличествоЗамен]])

ЗАМЕНИТЬ(СтрокаДляПоиска, ИскомаяСтрока, Замена [, НомерФрагмента [, КоличествоЗамен]])

Возвращает строку, состоящую из СтрокиДляПоиска, в которой все вхождения (или указанное количество) ИскомойСтроки заменены на Замену. Также есть возможность указать номер фрагмента, с которого необходимо осуществлять замену, и количество осуществляемых замен. Например, первое вхождение можно пропустить, а начать заменять со второго и осуществить только две замены. Поиск вхождения осуществляется с учетом регистра.

Параметры.

СтрокаДляПоиска	- Строка, в которой будут произведены поиск и замена.
ИскомаяСтрока	- Строка, которую необходимо найти и заменить.
Замена	- Строка, на которую необходимо заменить найденную ИскомуюСтроку.
НомерФрагмента	- Число, номер найденного фрагмента, с которого необходимо начать замены. Если параметр не указан, то замены начнутся с первого вхождения.
КоличествоЗамен	- Число, количество осуществляемых замен. Если параметр не указан, то будут заменены все найденные вхождения.

Возвращаемое значение - Строка, полученная из исходной строки заменой требуемых вхождений.

STRTRANC(СтрокаДляПоиска, ИскомаяСтрока, Замена [, НомерФрагмента [, КоличествоЗамен]])

Возвращает строку, состоящую из СтрокиДляПоиска, в которой все вхождения (или указанное количество) ИскомойСтроки заменены на Замену. Также есть возможность указать номер фрагмента, с которого необходимо осуществлять замену, и количество осуществляемых замен. Например, первое вхождение можно пропустить, а начать заменять со второго и осуществить только две замены. Поиск вхождения осуществляется без учета регистра.

Параметры.

СтрокаДляПоиска	- Строка, в которой будут произведены поиск и замена.
ИскомаяСтрока	- Строка, которую необходимо найти и заменить.
Замена	- Строка, на которую необходимо заменить найденную ИскомуюСтроку.
НомерФрагмента	- Число, номер найденного фрагмента, с которого необходимо начать замены. Если параметр не указан, то замены начнутся с первого вхождения.
КоличествоЗамен	- Число, количество осуществляемых замен. Если параметр не указан, то будут заменены все найденные вхождения.

Возвращаемое значение - Строка, полученная из исходной строки заменой требуемых вхождений.

SUBSTR(Строка, НачальнаяПозиция[, КоличествоСимволов])

ПОДСТРОКА(Строка, НачальнаяПозиция[, КоличествоСимволов])

Возвращает указанное количество символов строки, начиная с НачальнойПозиции. Если КоличествоСимволов меньше или равно нулю, то будет возвращена пустая строка. Если КоличествоСимволов не указано, то будет возвращена подстрока, начиная с НачальнойПозиции до конца исходной строки.

Параметры.

Строка	- Строка, из которой производится выборка подстроки.
НачальнаяПозиция	- Число, индекс в строке, с которого начинается выборка подстроки. Нумерация начинается с 1.
КоличествоСимволов	- Число, количество символов, которое необходимо вернуть.

Возвращаемое значение - Строка, состоящая из указанного количества символов исходной строки начиная с НачальнойПозиции.

UPPER(Строка)

Преобразует указанную строку в прописные (заглавные) символы.

Параметры.

Строка	- Строка, которая будет преобразована в прописные символы.
--------	--

Возвращаемое значение - Строка, полученная из исходной заменой всех символов на прописные.

VAL(Строка)

Преобразует указанную строку в числовое значение.

Параметры.

Строка - Строка, которая будет преобразована в число.

Возвращаемое значение - Число. Если в исходной строке есть десятичная точка или запятая, то результатом будет Число вещественное. Иначе - Число целое.

ПРОПИСЬ(Число, Слово1, Слово2, Слово5 [, КоличествоЗнаков [, МужЖенСредРод]])

Возвращает указанное число прописью и добавляет одно из указанных Слов в конце строки. Слова должны соответствовать формам, используемым с числительными, оканчивающимся, соответственно, на 1, кроме 11, на 2, 3 или 4, кроме 12,13,14 и на 5,6,7,8,9,11,12,13,14. Если указан параметр КоличествоЗнаков, то после Слов будет выведена дробная часть в виде цифр. Для корректного вывода числительных один, одно, одна, два и две необходимо указать параметр МужЖенСредРод.

Параметры.

Число - Число, которое будет написано прописью.

Слово1 - Строка, именительный падеж единственного числа Слова. Будет добавлено к числительным, оканчивающимся на 1, кроме 11.

Слово2 - Строка, родительный падеж единственного числа Слова. Будет добавлено к числительным, оканчивающимся на 2, 3 или 4, кроме 12, 13 и 14.

Слово5 - Строка, родительный падеж множественного числа Слова. Будет добавлено к числительным, оканчивающимся на 5, 6, 7, 8, 9, 11, 12, 13, 14 и 15.

КоличествоЗнаков - Число, количество знаков после запятой, которое будет выведено после Слов в виде цифр.

МужЖенСредРод - Число, указывает в каком роде выводить числительные один и два. 0- мужской род, 1- женский, 2- средний.

Возвращаемое значение – Строка, содержащая указанное число прописью.

Пример. ПРОПИСЬ(1234.56, 'стол', 'стола', 'столов', 2, 1) в качестве результата вернет «одна тысяча двести тридцать четыре стола 56».

ПРОПИСЬРУБ(Число[, КоличествоЗнаков])

Возвращает указанное количество рублей прописью. Функция является упрощенной формой функции ПРОПИСЬ.

Параметры.

Число	- Число, которое будет написано прописью. После числа прописью будет добавлено одно из слов 'рубль', 'рубля' или 'рублей'.
КоличествоЗнаков	- Число, количество копеек, которое будет выведено после Слов в виде цифр и слова 'копейка', 'копейки' или 'копеек'.

Возвращаемое значение – Строка, содержащая указанное количество рублей прописью.

Пример. ПРОПИСЬРУБ(1234.56, 2) в качестве результата вернет «одна тысяча двести тридцать четыре рубля 56 копеек».

Примеры

А. В данном примере иллюстрируется возможность применения функции Strtran для изменения формата вывода списка значений параметра отчета. В отчете был создан параметр Склад, в качестве типа параметра указан Список, в качестве источника – Справочник складов. Функция Strtran заменяет строки, состоящие из одинарных кавычек, на пустые строки.

```
Сообщение(Склад_список); -- выведет '001','002','003'  
Сообщение(Strtran(Склад_список, "'", "")); -- выведет 001,002,003
```

В. В данном примере рассматривается один из способов получения значения веса либо объема товара. Поля weight, ed_weight, d_litr и edd_litr относятся к некоторому товару, причем, чтобы в итоговом выражении не было лишних пробелов, к ним применяется функция AllTrim. Для этого weight и d_litr, которые являются числами, предварительно конвертируются в строки функцией Str. Переменные Вес и Объем не являются пустыми тогда и только тогда, когда вес и объем соответственно указаны в карточке товара, т.е. weight и d_litr не равны 0. Переменной ТоварОбъемВес присваивается значение единственной непустой переменной из этих двух.

```
Вес= ЕСЛИ(weight<>0, ALLTRIM(STR(weight,12,3))+""+ALLTRIM(ed_weight),  
"");  
Объем= ЕСЛИ(d_litr<>0, ALLTRIM(STR(d_litr,10,3))+""+ALLTRIM(edd_litr),  
"");  
ТоварОбъемВес= ЕСЛИ(!ПУСТО(Вес), Вес, Объем);
```

С. Данный пример является выражением по умолчанию для отображения на дисплее прайс-чекера. функцией AllTrim убираются лишние пробелы у строковой переменной Наименование, функцией Str числовая переменная цена конвертируется в строку. Затем функцией PAdr строковые выражения дополняются пробелами справа до размера в сорок символов.

```
PADR(ALLTRIM(НАИМЕНОВАНИЕ), 40) + PADR('ЦЕНА '+STR(ЦЕНА, 10, 2), 40)
```

Функции даты/времени

CDOW(Дата|Число)

ИМЯДНЯНЕДЕЛИ(Дата|Число)

Возвращает имя дня недели по указанной дате или по номеру дня недели.

Параметры.

Дата	- Дата, для которой рассчитывается имя дня недели.
НомерДня	- Число, номер дня недели, имя которого необходимо определить.

Возвращаемое значение - Строка, содержащая имя дня недели. Понедельник - первый день.

CONSTANTSDATE([Дата])

ДАТАКОНСТАНТ([Дата])

Устанавливает дату для определения значений периодических констант.

Параметры.

Дата	- Дата, на которую будут вычислены значения используемых констант. Если параметр не указан, то будет использована текущая дата.
------	---

Возвращаемое значение - нет.

CTOD(Строка|Число)

Преобразует указанную строку или число в дату.

Параметры.

Строка	- Строка, содержащая дату в формате ДД.ММ.ГГГГ ЧЧ::ММ:СС.
Число	- Номер дня по Юлианскому календарю, который надо преобразовать в дату.

Возвращаемое значение - ДатаВремя. Если указанная строка не может быть преобразована в дату, то будут возвращены пустые дата и время.

DATE()

ДАТА()

Возвращает текущую системную дату (без времени).

Параметры.

нет.

Возвращаемое значение - ДатаВремя, содержащее текущую системную дату без времени.

DATETIME()
ДАТАВРЕМЯ()

Возвращает текущую системную дату и время.

Параметры.
нет.

Возвращаемое значение - ДатаВремя, содержащее текущие системную дату и время.

DAY(Дата)
ДЕНЬ(Дата)

Возвращает номер дня месяца из указанной даты.

Параметры.
Дата - Дата, для которой рассчитывается номер дня месяца.

Возвращаемое значение - Число, соответствующее номеру дня месяца указанной даты.

DOW(Дата)
ДЕНЬНЕДЕЛИ(Дата)

Возвращает номер дня недели по указанной дате.

Параметры.
Дата - Дата, для которой рассчитывается номер дня недели.

Возвращаемое значение - Число, соответствующее номеру дня недели указанной даты.

DTOS(Дата[, ФорматДаты[, РазделительДаты]])

Преобразует выражение типа Дата в строку в указанном формате. Если формат преобразования не указан, то выбирается формат, соответствующий региональным установкам. Если не указан разделитель даты, то используется разделитель, указанный в региональных настройках.

Параметры.

Дата	- Дата, которая будет преобразована в строку.
ФорматДаты	- Число, формат преобразования даты. Допустимые значения от 0 до 7. 0 - ДД.ММ.ГГ 1 - ММ.ДД.ГГ 2 - ГГ.ДД.ММ 3 - ГГ.ММ.ДД 4 - ДД.ММ.ГГГГ 5 - ММ.ДД.ГГГГ 6 - ГГГГ.ДД.ММ 7 - ГГГГ.ММ.ДД
РазделительДаты	- Символ-разделитель. Если параметр не указан, будет использован разделитель, указанный в региональных настройках системы.

Возвращаемое значение - строка, содержащая указанную дату.

GODAY(Дата, [КоличествоДней= 1])

Возвращает дату, отстоящую от указанной даты на указанное количество дней. Если КоличествоДней больше нуля, то полученная дата будет позже исходной. Если КоличествоДней меньше нуля, то полученная дата будет раньше исходной.

Параметры.

Дата	- Дата, от которой производится расчет.
КоличествоДней	- Число, количество дней, которое необходимо прибавить к указанной дате. Если параметр не указан, то прибавляется 1 день.

Возвращаемое значение - Дата, отстоящая от указанной на указанное количество дней.

GOMONTH(Дата, [КоличествоМесяцев= 1])

Возвращает дату, отстоящую от указанной даты на указанное количество месяцев. Если КоличествоМесяцев больше нуля, то полученная дата будет позже исходной. Если КоличествоМесяцев меньше нуля, то полученная дата будет раньше исходной.

Параметры.

Дата	- Дата, от которой производится расчет.
КоличествоМесяцев	- Число, количество месяцев, которое необходимо прибавить к указанной дате. Если параметр не указан, то прибавляется 1 месяц.

Возвращаемое значение - Дата, отстоящая от указанной на указанное количество месяцев.

GOYEAR(Дата, [КоличествоЛет= 1])

Возвращает дату, отстоящую от указанной даты на указанное количество лет. Если КоличествоЛет больше нуля, то полученная дата будет позже исходной. Если КоличествоЛет меньше нуля, то полученная дата будет раньше исходной.

Параметры.

Дата	- Дата, от которой производится расчет.
КоличествоЛет	- Число, количество лет, которое необходимо прибавить к указанной дате. Если параметр не указан, то прибавляется 1 год.

Возвращаемое значение - Дата, отстоящая от указанной на указанное количество лет.

HOUR(ДатаВремя)

Возвращает номер часа из указанных даты и времени.

Параметры.

ДатаВремя	- Дата и время, для которых функция вернет значение.
-----------	--

Возвращаемое значение - Число целое от 0 до 23, номер часа из указанных даты и времени.

MINUTE(ДатаВремя)

МИНУТА(ДатаВремя)

Возвращает значение минут из указанных даты и времени.

Параметры.

ДатаВремя	- Дата и время, для которых функция вернет значение.
-----------	--

Возвращаемое значение - Число целое от 0 до 59, значение минут из указанных даты и времени.

MONTH(Дата)
МЕСЯЦ(Дата)

Возвращает номер месяца из указанной даты.

Параметры.

Дата - Дата, для которой функция вернет значение.

Возвращаемое значение - Число целое от 1 до 12, номер месяца из указанной даты.

MONTHFIRSTDATE(Дата)
НАЧАЛОМЕСЯЦА(Дата)

Возвращает дату, соответствующую началу месяца для указанной даты.

Параметры.

Дата - Дата, для которой рассчитывается дата начала месяца.

Возвращаемое значение - Дата, соответствующая началу месяца для указанной даты.

MONTHLASTDATE(Дата)
КОНЕЦМЕСЯЦА(Дата)

Возвращает дату, соответствующую последнему дню месяца для указанной даты.

Параметры.

Дата - Дата, для которой рассчитывается последний день месяца.

Возвращаемое значение - Дата, соответствующая последнему дню месяца для указанной даты.

MONTHNAME(Дата [,НомерМесяца [,Параметр2]])
ИМЯМЕСЯЦА(Дата [,НомерМесяца [,Параметр2]])

Возвращает название месяца из указанной даты или по указанному номеру месяца.

Параметры.

Дата - Дата, для которой функция вернет значение.

НомерМесяца - Номер месяца от 1 до 12, для которого функция вернет значение.

Параметр2 - Число. Если значение параметра равно нулю, то будет возвращено название месяца в Именительном падеже. Иначе - в Родительном падеже. Если параметр не указан, то используется значение 0.

Возвращаемое значение - Строка, содержащая название месяца в указанном падеже.

QUARTER(Дата)
КВАРТАЛ(Дата)

Возвращает номер квартала в году по указанной дате.

Параметры.

Дата - Дата, для которой рассчитывается номер квартала.

Возвращаемое значение - Число, соответствующее номеру квартала в году для указанной даты.

TIME()
ВРЕМЯ()

Возвращает текущее системное время.

Параметры.

нет.

Возвращаемое значение - ДатаВремя, содержащее текущее системное время. Поле Дата возвращаемого значения содержит 0.

TTOS(ДатаВремя[, ФорматДаты[, РазделительДаты]])

Преобразует выражение типа ДатаВремя в строку в указанном формате. Если формат преобразования не указан, то выбирается формат, соответствующий региональным установкам. Если не указан разделитель даты, то используется разделитель, указанный в региональных настройках.

Параметры.

ДатаВремя	- Дата и время, которые необходимо преобразовать в строку.
ФорматДаты	- Число, формат преобразования даты. Допустимые значения от 0 до 8. 0 - ДД.ММ.ГГ ЧЧ:ММ:СС 1 - ММ.ДД.ГГ ЧЧ:ММ:СС 2 - ГГ.ДД.ММ ЧЧ:ММ:СС 3 - ГГ.ММ.ДД ЧЧ:ММ:СС 4 - ДД.ММ.ГГГГ ЧЧ:ММ:СС 5 - ММ.ДД.ГГГГ ЧЧ:ММ:СС 6 - ГГГГ.ДД.ММ ЧЧ:ММ:СС 7 - ГГГГ.ММ.ДД ЧЧ:ММ:СС 8 - ЧЧ:ММ:СС, только время
РазделительДаты	- Символ-разделитель. Если параметр не указан, будет использован разделитель, указанный в региональных настройках системы.

Возвращаемое значение - строка, содержащая указанные дату и время.

TTOD(ДатаВремя)

Преобразует указанные дату и время в дату без времени.

Параметры.

ДатаВремя - Дата и время, которые необходимо преобразовать в Дату.

Возвращаемое значение - ДатаВремя, полученное из исходного значения, но с полем Время, равным 0.

WEEK(Дата)

НЕДЕЛЯ(Дата)

Возвращает номер недели в году по указанной дате.

Параметры.

Дата - Дата, для которой рассчитывается номер недели.

Возвращаемое значение - Число, соответствующее номеру недели в году для указанной даты.

YEAR(Дата)

ГОД(Дата)

Возвращает год из указанной даты.

Параметры.

Дата - Дата, для которой возвращается год.

Возвращаемое значение - Число, соответствующее номеру года для указанной даты.

Примеры

А. В данном примере переменная типа ДатаВремя формируется на основании параметра, содержащего год в числовом виде. Функцией STR год преобразуется в строку и прибавляется к другой строке, содержащей день и месяц. Полученная строка функцией CTOD конвертируется в переменную типа ДатаВремя.

```
Период_min=CTOD("01.01."+STR(Год_отчета,4));
```


В. В данном примере параметр отчета Интервал_min типа ДатаВремя интерпретируется функциями DTOC и TTOC как содержащий только дату или и дату, и время соответственно. Полученные в результате преобразования строки подставляются в запрос, в котором средствами самого SQL они конвертируются в тип datetime и полученные значения присваиваются переменным.

```
Запрос("
DECLARE @p_min_d datetime, @p_min_dt datetime;
SET @p_min_d=CONVERT(datetime, '"+DTOC(Интервал_min)+'');
SET @p_min_dt=CONVERT(datetime, '"+TTOC(Интервал_min)+'');
");
```

Прочие функции

BETWEEN(Выражение1, Выражение2, Выражение3)

МЕЖДУ(Выражение1, Выражение2, Выражение3)

Вычисляет, находится ли Выражение1 между Выражением2 и Выражением3

Параметры.

Выражение1 - Число, Строка или ДатаВремя.

Выражение2 - Выражение, тип которого должен совпадать с типом Выражения1.

Выражение3 - Выражение, тип которого должен совпадать с типом Выражения1

Возвращаемое значение - Логическое. Истина, если выполняется неравенство
Выражение2 ≤ Выражение1 ≤ Выражение3. Иначе - Ложь.

EMPTY(Выражение)

ПУСТО(Выражение)

Возвращает ИСТИНУ, если указанное выражение пустое. Строка считается пустой, если она не содержит символов или коды всех символов не превосходят 32 (код пробела). Число считается пустым, если оно равно 0. Дата считается пустой, если ее числовое представление равно 0. Логическое выражение считается пустым, если его значение равно ЛОЖЬ.

Параметры.

Выражение - Выражение, значение которого будет проанализировано.

Возвращаемое значение - Логическое. Истина, если значение выражения считается пустым.
Иначе - Ложь.

IIF(Условие, Выражение1, Выражение2)
ЕСЛИ(Условие, Выражение1, Выражение2)

Функция возвращает одно из двух значений, в зависимости от истинности указанного Условия. Если Условие истинно, то возвращается значение Выражения1. Иначе - значение Выражения2.

Параметры.

- | | |
|------------|---|
| Условие | - Логическое выражение, значение которого используется при выборе возвращаемого значения. |
| Выражение1 | - Выражение любого типа, значение которого будет возвращено, если Условие истинно. |
| Выражение2 | - Выражение любого типа, значение которого будет возвращено, если Условие ложно. |

Возвращаемое значение - зависит от значения Условия и указанных Выражений.

INLIST(Выражение1,Выражение2,...)
ВСПИСКЕ(Выражение1,Выражение2,...)

Проверяет наличие указанного значения в списке значений. Размер указанного списка значений неограничен.

Параметры.

- | | |
|----------------|---|
| Выражение1 | - Выражение любого типа, значение которого будет искаться в списке. |
| Выражение2,... | - Список выражений, среди значений которых будет осуществлен поиск. |

Возвращаемое значение - Логическое. Истина, если значение Выражения1 находится в списке значений Выражение2,...,ВыражениеN.

MAX(Выражение1, Выражение2)

Возвращает максимальное из двух значений.

Параметры.

- | | |
|------------|--|
| Выражение1 | - Выражение любого типа, значение которого будет участвовать в сравнении. |
| Выражение2 | - Выражение того же типа, что и Выражение1, значение которого будет участвовать в сравнении. |

Возвращаемое значение - максимальное из значений указанных выражений.

MIN(Выражение1, Выражение2)

Возвращает минимальное из двух значений.

Параметры.

- | | |
|------------|--|
| Выражение1 | - Выражение любого типа, значение которого будет участвовать в сравнении. |
| Выражение2 | - Выражение того же типа, что и Выражение1, значение которого будет участвовать в сравнении. |

Возвращаемое значение - минимальное из значений указанных выражений.

RAND([Параметр])

СЛУЧАЙНОЕ ЧИСЛО([Параметр])

Возвращает псевдослучайное число. Если Параметр не указан, то будет возвращено следующее псевдослучайное число. Если в качестве параметра передано число, то генератор псевдослучайных чисел будет инициализирован этим числом. Если передан параметр любого другого типа, например, логический, то генератор псевдослучайных чисел будет инициализирован текущими системными датой и временем.

Параметры.

- | | |
|----------|---|
| Параметр | - Число, база для инициализации генератора псевдослучайных чисел. |
|----------|---|

Возвращаемое значение - Число вещественное, псевдослучайное число.

TYPE(Выражение)

ТИП(Выражение)

Возвращает тип значения указанного выражения.

Параметры.

- | | |
|-----------|---|
| Выражение | - Выражение, значение которого необходимо определить. Так как в системе отсутствует строгая типизация данных, для определения типа Выражения необходимо его вычислить, и по результату вычисления определить тип. |
|-----------|---|

Возвращаемое значение - Строка. В зависимости от определенного типа Выражения, возвращается один из символов. Все символы заглавные латинские.

'U' - выражение не может быть вычислено.

'I' - тип результата вычисления Выражения: Число целое.

'B' - тип результата вычисления Выражения: Число вещественное.

'C' - тип результата вычисления Выражения: Строка.

'L' - тип результата вычисления Выражения: Логический

'D' - тип результата вычисления Выражения: ДатаВремя.

UNIQUENAME() УНИКАЛЬНОЕИМЯ()

Функция возвращает строку, которая может быть использована в качестве уникального имени. Данное имя может быть использовано, например, для создания временных таблиц.

Параметры.
нет.

Возвращаемое значение - Строка, содержащая уникальную последовательность символов.

Пример

В данном примере функция УникальноеИмя используется для генерации имени временной таблицы. С помощью функции Empty осуществляется проверка того, что при формировании отчета в параметрах был выбран склад. В случае выполнения данного условия, функция IIF добавит к запросу дополнительную строку с условием, иначе она добавит к нему пустую строку, ничего тем самым не меняя.

```
ИмяТаблицы1= "#"+УникальноеИмя();
```

```
Запрос("
select a.code, a.name, a.clev, a.branch,
cast(isnull(b.dbase, '') as varchar(30)) as dbase, isnull(c.name,
'себестоимость') as name_clev
into "+ИмяТаблицы4+" from sprskl a
left outer join sprbranch b on b.code=a.branch and b.folder=0
left outer join sprccnames c on c.clev=a.clev"+
IIF(!Empty(Склад), " where a.code in ("Склад_список+)", "")
);
```

Управляющие конструкции

BREAK выход из последнего цикла WHILE

Прерывает выполнение цикла WHILE и передает управление оператору, следующему за блоком команд цикла WHILE.

Синтаксис
BREAK

IF...ELSE условный оператор

Выполняет блок команд при выполнении указанного Условия или альтернативный блок команд при невыполнении Условия.

Синтаксис

IF (Условие)

{ блок команд }

[

ELSE

{ блок команд }

]

Условный оператор IF может быть многократно вложен один в другой, либо в другие блоки команд.

RETURN выход из процедуры

Выполняет возврат из выполняемой процедуры или функции.

Синтаксис

RETURN (Возвращаемое значение)

При выходе из функции Возвращаемое значение будет являться значением этой функции.

WHILE цикл по условию

Выполняет блок команд циклически, пока выполняется указанное Условие.

Синтаксис

WHILE (Условие)

{

блок команд

[BREAK]

}

Команды внутри блока команд выполняются циклически, пока выполняется указанное Условие.

Циклы по условию могут быть вложены один в другой, либо в другие блоки команд. Работа цикла может быть прервана посредством оператора BREAK.

Примеры

- A. В данном примере объявляется переменная-счетчик *i*, инициализируемая значением 1. В теле бесконечного цикла WHILE происходит увеличение счетчика и с помощью оператора IF осуществляется проверка выполнения условия “*i* равняется 10”. В случае, если условие выполнено, значение счетчика уменьшается на 3 и с помощью оператора BREAK выполнение цикла прерывается. Таким образом, в результате данных действий значение *i* установится в 7.

```
i=1;
WHILE (Истина)
{
    i++;
    IF (i==10)
    {
        i=i-3;
        BREAK;
    }
}
```

- B. В данном примере содержится код, исполняемый при входе в отчет. Если Empty(Учет) возвращает истину, пользователю выдается сообщение и на этом путем вызова функции Return формирование отчета прекращается.

```
IF (Empty(Учет))
{
    Сообщение("Не выбран учет!");
    Return -1;
}
```