

Оглавление

[Структура PL/SQL блока](#)

[Раздел объявлений](#)

[Раздел выполнения](#)

[Раздел обработки исключительных ситуаций](#)

[Анонимные блоки](#)

[Синтаксис языка](#)

[Идентификаторы](#)

[Арифметические операторы:](#)

[Операторы отношения \(используются в логических выражениях\):](#)

[Комментарии и метки](#)

[Основные типы и структуры данных](#)

[Типы LOB](#)

[Составные типы](#)

[Динамические типы](#)

[Пользовательские подтипы](#)

[Объявление переменных и постоянных](#)

[Функции для работы с NULL](#)

[DECODE](#)

[NVL](#)

[NVL2](#)

[COALESCE](#)

[Неявное преобразование типов](#)

[Явное преобразование типов](#)

[Основные управляющие структуры](#)

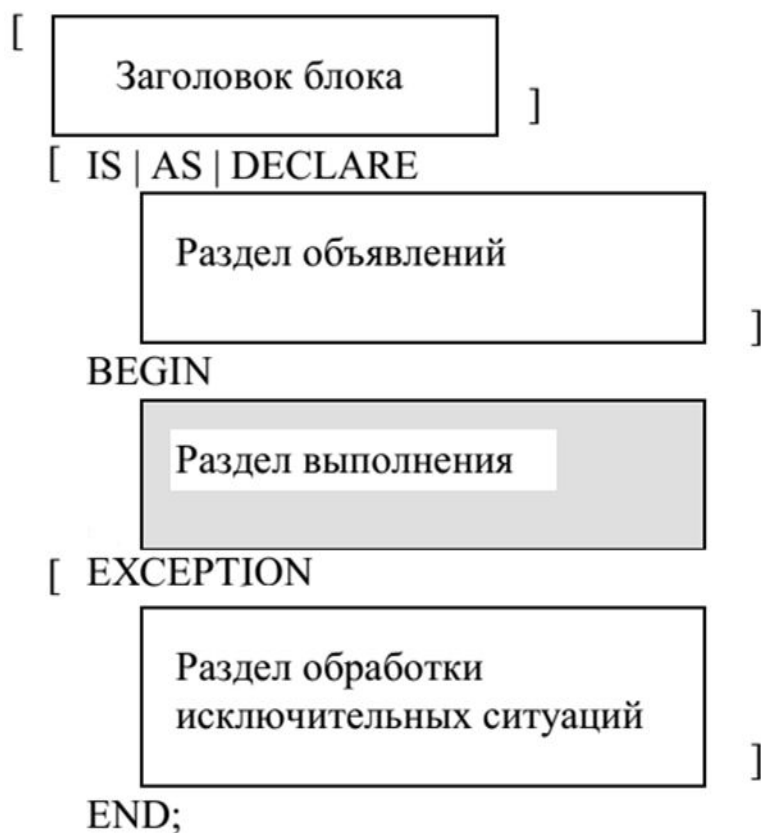
[Выбор по условию](#)

[Циклы](#)

[Goto \(безусловный переход\)](#)

[Метки в циклах и блоках](#)

Структура PL/SQL блока



Все разделы, кроме раздела выполнения, имеют право отсутствовать.

Раздел выполнения может наряду с программным кодом содержать произвольное число других (вложенных) блоков PL/SQL.

Раздел объявлений

Содержит объявления переменных, констант, курсоров, исключений, функций и процедур PL/SQL, которые будут использоваться в выполняемой секции и секции исключений

Все, что находится в секции объявлений, принадлежит блоку и может использоваться только внутри него.

Раздел выполнения

Выполняемая секция начинается с ключевого слова BEGIN и заканчивается либо ключевым словом EXCEPTION, если присутствует раздел исключений, либо ключевым словом END.

Раздел обработки исключительных ситуаций

Раздел исключений начинается с ключевого слова EXCEPTION и продолжается до конца блока. Каждому исключению соответствует оператор WHEN имя_исключения,

указывающий, что должно быть сделано при возникновении данного исключения.

Все операторы, находящиеся между оператором, вызвавшим ошибку, и секцией исключений, игнорируются.

Анонимные блоки

Анонимный блок (anonymous block) — это блок PL/SQL без секции заголовка

Синтаксис языка

Идентификаторы

К идентификаторам PL/SQL предъявляются следующие требования:

- Иметь не более 30 символов в длину и не содержать пробельных символов (собственно пробелов и знаков табуляции).
- Состоять только из букв, цифр от 0 до 9, символа подчеркивания (_), знака доллара (\$) и знака фунта (#).
- Начинаться с буквы.
- Не совпадать с зарезервированными словами PL/SQL или SQL, которые имеют специальное значение. Например, именем переменной не может быть слово BEGIN или INSERT. Все зарезервированные слова можно увидеть в представлении v\$reserved_words.

:=	Присвоение
(Начало списка или подвыражения
)	Конец списка или подвыражения
,	Отдельные элементы списка (как в списке параметров)
..	Оператор диапазона используется в операторах FOR-IN
 	Конкатенация строк
= >	Ассоциация (используется в списке параметров)
;	Конец выражения
%	Атрибут курсора или типа объекта
.	Спецификация объекта
@	Индикатор удаленной базы данных
,	Начало/конец строки символов
:	Индикатор внешней переменной
&	Индикатор связанной переменной

Арифметические операторы:

+	Сложение и унарный плюс
-	Вычитание и унарный минус
*	Умножение
/	Деление
* *	Возведение в степень

Операторы отношения (используются в логических выражениях):

=	Равенство
<	Меньше
>	Больше
< >	Не равно
!=	Не равно (альтернатива)
~ =	Не равно (альтернатива)
^ =	Не равно (альтернатива)
< = >	Меньше или равно
> = <	Больше или равно

Комментарии и метки

--	Комментарий в одной строке
/*	Начало многострочного комментария
*/	Конец многострочного комментария
> >	Начало метки
< <	Конец метки

Основные типы и структуры данных

Вид данных	Описание
------------	----------

С к а л я р н ы й	П е р е м е н н ы е , п р е д с т а в л я ю щ и е с о б о й р о в н о о д н о з н а ч е н и е (ч и с л о в о е , д а т у и т . д .)
С о с т а в н о й	П е р е м е н н ы е , п р е д с т а в л я ю щ и е и м е н о в а н н у ю г р у п п у з н а ч е н и й (з а п и с ь , о б ь е к т , м а с с и в)
С с ы л к а	С с ы л к а н а о б ь е к т и л и к у р с о р
LOB	У к а з а н и е н а м а с с и в б о л ь ш о г о р а з м е р а

Null

Это специальное значение – означает отсутствие данных, констатацию того факта, что значение неизвестно. По умолчанию это значение могут принимать переменные всех типов данных, если явно не указано ограничение NOT NULL, если ограничение указано тогда надо определять значение по умолчанию. Основная особенность заключается в том, что Null ни равен ничему, даже другому Null.

Для сравнения существует специальное сравнение «is», с помощью которого мы можем сравнивать разные переменные.

Таким образом, Oracle оперирует не двухзначной, а трехзначной логикой

Скалярные типы

Числовые типы

Data Type	Data Description
PLS_INTEGER or BINARY_INTEGER (с 10й версии одно и тоже)	Знаковое целое диапазон значений -2 147 483 648 до 2 147 483 647, размещается в 32 битах
BINARY_FLOAT	Числа с одинарной точностью, соответствует формату IEEE 754-формат с плавающей запятой (32 бита, от $\pm 2^{-149}$ до $\pm 2^{127} \cdot (2 \cdot 2^{-23})$)
BINARY_DOUBLE	Числа двойной точности, соответствует формату IEEE 754-формат с плавающей запятой (64 бита, диапазон от $\pm 2^{-1074}$ до $\pm 2^{1023} \cdot (2 \cdot 2^{-52})$)
NUMBER [(p [, s])]	С фиксированной или плавающей запятой с абсолютным значением в диапазоне от 1E-130 до (но не включая) 1.0E126. Может содержать ноль. Precision – общее число значащих цифр (max – 38) Scale – количество цифр справа от запятой (от -84 до 127)

Подтипы PLS_INTEGER:

Data Type	Data Description
NATURAL	Неотрицательное PLS_INTEGER
NATURALN	Неотрицательное PLS_INTEGER значение с NOT NULL ограничением
POSITIVE	Положительное PLS_INTEGER значение (начинается с 1)
POSITIVEN	Положительное PLS_INTEGER значение с NOT NULL ограничением
SIGNTYPE	PLS_INTEGER значение -1, 0, или 1 (полезно при программировании tri-state логики (три состояния))
SIMPLE_INTEGER	PLS_INTEGER значение с NOT NULL ограничением

Подтипы BINARY_FLOAT/DOUBLE:

Data Type	Data Description
SIMPLE_FLOAT	BINARY_FLOAT значение с NOT NULL ограничением
SIMPLE_DOUBLE	BINARY_DOUBLE значение с NOT NULL ограничением

Строковые типы

Data Type	Data Description
CHAR [(size [BYTE CHAR])]	Строки фиксированной длины до 32767 байт (в Oracle SQL предел в 2000 байт)
VARCHAR2(size [BYTE CHAR])	Строки переменной длины до 32767 байт (в Oracle SQL предел в 4000 байт)
NCHAR[(size)] и NVARCHAR2(size)	NLS-символьные типы. Позволяют обрабатывать символьные данные в мультисимвольной кодировке Unicode
RAW(size)	Байтовая строка переменной длины до 32767 байт (в Oracle SQL 2000 байт)
LONG	Строки переменной длины до 32767 байт. Тип сохранен для обратной совместимости; в частности встречается в некоторых справочных таблицах
LONG RAW	Байтовая строка переменной длины до 32767 байт. Тип сохранен для обратной совместимости версий Oracle

Следующие типы формально можно причислить к строковым, но используются они для представления физических адресов:

Data Type	Data Description
ROWID	Двоичный массив фиксированной длины для хранения физического адреса данных Oracle в шестнадцатеричном формате OOOOOOFFFFBBBBBBBBRRR
UROWID [(size)]	«Универсальный» формат для ROWID: шестнадцатеричная строка переменной длины

	(до 4000 байт) с логическим значением ROWID. Используется для хранения адресов строк в индексно организованных (index organized) таблицах или в таблицах DB2 (через шлюз)
--	--

В Oracle строковая переменная равная "" (пустая строка) эквивалентна null.

Типы для моментов и интервалов времени

Data Type	Data Description
DATE	Допустимый диапазон дат от 1 января 4712 г. до н.э., до 31 декабря 9999 года нашей эры. Формат по умолчанию определяется явно в параметре NLS_DATE_FORMAT или неявно в NLS_TERRITORY. Размер фиксируется в 7 байт. Этот тип данных содержит поля даты (Год, месяц, день, час, минуту и секунду. Не содержит дробной части секунд и часового пояса
TIMESTAMP [(fractional_seconds_precision)] [WITH TIME ZONE WITH LOCAL TIME ZONE]	То же что и дата + секунды имеют дробный формат. fractional_seconds_precision [0;9] – количество цифр в дробной части. По умолчанию 6. Может содержать часовой пояс или указать сразу локальный
INTERVAL YEAR [(year_precision)] TO MONTH	Хранит период времени в годах и месяцах. - year_precision - это количество цифр в YEAR. Допустимые значения от 0 до 9. По умолчанию 2.
INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	Хранит период времени в днях и секундах. - day_precision – это количество цифр в DAY. Допустимые значения от 0 до 9. По умолчанию 2. - fractional_seconds_precision – количество цифр в дробной части SECOND. Допустимые значения от 0 до 9. Значение по умолчанию 6.

Поддерживаемые значения:

Поле	Диапазоны значения для Datetime	Диапазоны для интервальных значений
YEAR	-4712 to 9999 (исключая 0)	Любое ненулевое целое
MONTH	01 до 12	0 до 11
DAY	01 до 31 (ограничено значениями MONTH и YEAR, согласно локальным правилам календаря)	Любое ненулевое целое
HOUR	00 до 23	0 до 23
MINUTE	00 до 59	0 до 59

SECOND	00 до 59.9(n), где 9(n) это установка fractional_seconds_precision	0 до 59.9(n), где 9(n) это установка fractional_seconds_precision
TIMEZONE_HOUR	-12 до 14 (учитывает переход на летнее время)	Не используется
TIMEZONE_MINUTE	00 до 59	Не используется
TIMEZONE_REGION	Выбирается в представлении V\$TIMEZONE_NAMES	Не используется
TIMEZONE_ABBR	Выбирается в представлении V\$TIMEZONE_NAMES	Не используется

Операции над типом и результат

Operand 1	Operator	Operand 2	Result Type
datetime	+	interval	datetime
datetime	-	interval	datetime
interval	+	datetime	datetime
datetime	-	datetime	interval
interval	+	interval	interval
interval	-	interval	interval
interval	*	numeric	interval
numeric	*	interval	interval
interval	/	numeric	interval

Булевы типы

Фактически – тип для трехзначных переменных с допустимыми значениями TRUE, FALSE и NULL.

В выражениях допустимы только булевы операнды.

Типы LOB

Large Objects «большие неструктурированные объекты»

Data Type	Data Description
BFILE	Указатель на файл с данными в операционной системе (Не больше 4 Гб)
BLOB	Указатель на большой неструктурированный массив в БД (до 128 терабайт)
CLOB	Указатель на большой символьный массив в БД (до 128 терабайт)
NCLOB	Указатель на большой символьный массив в многобайтовой кодировке (до 128 терабайт)

Составные типы

Записи являются примером данных составного типа. В PL/SQL они бывают трех видов:

- воспроизводящими структуру таблицы в БД
- воспроизводящими структуру курсора в программе

- заданными пользователем произвольно

Помимо этого к числу составных типов (допускающих структуру значения) Oracle относит типы объектов и коллекций.

Записи в PL/SQL могут объявляться в разделе объявлений блока или в разделе глобальных описаний пакета.

CREATE TYPE имя_типа IS RECORD (объявление переменных через “,”);

Записи, повторяющие структуру таблицы или курсора, объявляются с помощью атрибута %ROWTYPE.

Записи, задаваемые пользователем, объявляются через предложение TYPE. Из примера видно, что записи могут быть вложенными.

Динамические типы

%TYPE	Ссылка на тип поля в столбце (v_id customer.id%type;)
%ROWTYPE	Ссылка на структуру таблицы, создается запись (Record) (v_rec_customer%rowtype

Пользовательские подтипы

SUBTYPE имя_подтипа IS базовый_тип [(ограничения)] [NOT NULL];

Где

- базовый_тип – любой скалярный или пользовательский PL/SQL тип данных.
- Ограничения – здесь можно указать точность, размер, масштаб и т.д.

Объявление переменных и постоянных

Общий формат объявления выглядит так:

имя_переменной [CONSTANT] тип_данных [NOT NULL] [{:= | DEFAULT} выражение];

CONSTANT – переменная будет константой.

DEFAULT(:=) задает значение по умолчанию (можно ссылаться на вышеописанные переменные, с уже установленным значением по умолчанию).

Функции для работы с NULL

DECODE

Ищет первое совпадение expr и search и возвращает result в случае успеха, иначе возвращает default или null



NVL2

COALESCE

Неявное преобразование типов

Почему нельзя использовать:

- Операторы SQL легче понять, когда Вы используете функции явного преобразования типа данных.
- Неявное преобразование типа данных может оказать отрицательное влияние на производительность, особенно если тип данных столбца преобразуется к типу данных константы, а не наоборот.
- Неявное преобразование зависит от контекста, в котором оно происходит, возможно, не будет работать одинаково в каждом случае. Например, неявное преобразование значения типа данных VARCHAR2 может вернуть неожиданный год в зависимости от значения параметра NLS_DATE_FORMAT.
- Алгоритмы для неявного преобразования подвержены изменениям при обновлении версий продуктов Oracle. Поведение явных преобразований более предсказуемо.

Таблица неявных преобразований в выражениях

COLUMN	NUMBER	FLOAT	DOUBLE	DATE	TS	TS_LTZ	TS_TZ	CHAR	RAW
NUMBER	NATIVE	LEFT	LEFT	ERROR	ERROR	ERROR	ERROR	RIGHT	ERROR
FLOAT	RIGHT	NATIVE	LEFT	ERROR	ERROR	ERROR	ERROR	RIGHT	ERROR
DOUBLE	RIGHT	RIGHT	NATIVE	ERROR	ERROR	ERROR	ERROR	RIGHT	ERROR

DATE	ERROR	ERROR	ERROR	NATIVE	LEFT	LEFT	LEFT	RIGHT	ERROR
TS	ERROR	ERROR	ERROR	RIGHT	NATIVE	RIGHT	LEFT	RIGHT	ERROR
TS_LTZ	ERROR	ERROR	ERROR	RIGHT	RIGHT	NATIVE	LEFT	RIGHT	ERROR
TS_TZ	ERROR	ERROR	ERROR	RIGHT	RIGHT	RIGHT	NATIVE	RIGHT	ERROR
CHAR	LEFT	LEFT	LEFT	LEFT	LEFT	LEFT	LEFT	NATIVE	RIGHT
RAW	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	LEFT	NATIVE

NATIVE	Не конвертируется. «Native» (тип поля = типу литерала)
RIGHT	«Light» конвертация применяется к LITERAL, к типу поля или переменной (COLUMN): COLUMN = to_column_type(LITERAL).
LEFT	«Heavy» конвертация поля или переменной (COLUMN) к типу LITERAL type: to_literal_type(COLUMN) = LITERAL.
ERROR	Неявное преобразование невозможно.

К примеру при сравнении NUMBER и CHAR неявному преобразованию подлежит правый операнд, т.е. CHAR

Явное преобразование типов

Функции явного преобразования типов

from	to	VARCHAR2 NVARCHAR2	NUMBER	Datetime Interval	RAW	CLOB, NCLOB, BLOB	BINARY_FLOAT BINARY_DOUBLE
VARCHAR2 NVARCHAR2		TO_CHAR (char.) TO_NCHAR (char.)	TO_NUMBER	TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ TO_YMINTERVAL TO_DSINTERVAL	HEXTORAW	TO_CLOB TO_NCLOB	TO_BINARY_FLOAT TO_BINARY_DOUBLE
NUMBER		TO_CHAR (number) TO_NCHAR (number)	--	TO_DATE NUMTOYM- INTERVAL NUMTODS- INTERVAL	--	--	TO_BINARY_FLOAT TO_BINARY_DOUBLE
Datetime Interval		TO_CHAR (date) TO_NCHAR (datetime)	--	--	--	--	--
RAW		RAWTOHEX RAWTONHEX	--	--	--	TO_BLOB	--
CLOB,		TO_CHAR TO_NCHAR	--	--	--	TO_CLOB	--

NCLOB, BLOB					TO_NCLOB	
CLOB, NCLOB, BLOB	TO_CHAR TO_NCHAR	--	--	--	TO_CLOB TO_NCLOB	--
BINARY_FLOAT BINARY_DOUBLE	TO_CHAR (char.) TO_NCHAR (char.)	TO_NUMBER	--	--	--	TO_BINARY_FLOAT TO_BINARY_DOUBLE

Основные управляющие структуры

Выбор по условию

Предложение IF-THEN

```
IF условие_выражение
THEN
    программный код в случае TRUE
END IF;
```

Предложение IF-THEN-ELSE

```
IF условие_выражение
THEN
    программный код в случае TRUE
ELSE
    программный код в случае FALSE/NULL
END IF;
```

Простой CASE

```
CASE выражение
WHEN выражение_для_сравнения1 THEN программный код1
[WHEN выражение_для_сравненияi THEN программный кодi]
...
[ELSE программный код]
END CASE;
```

CASE с поиском

```
CASE
WHEN условие_выражение1 THEN программный код1
```

[WHEN условное_выражение THEN программный код]

...

[ELSE программный код]

END CASE;

Циклы

[метка] LOOP

 Программный код

END LOOP [метка];

EXIT – прекращает выполнение цикла. WHEN – позволяет задать условие выхода

CONTINUE – завершает текущую итерацию цикла. WHEN – позволяет задать условие окончания итерации.

While

WHILE условное_выражение

LOOP

 программный код

END LOOP [слово-комментарий]

For по счетчику

FOR индекс_цикла IN [REVERSE] нижнее_значение ..
верхнее_значение

LOOP

 программный код

END LOOP [слово-комментарий]

индекс_цикла заводится в PL/SQL автоматически как переменная типа PLS_INTEGER, и объявлять его не требуется.

указание REVERSE заставить перевернет счетчик и он будет декрементировать от верхнее_значение до нижнее_значение

For по курсору

FOR индексная_запись IN [имя_курсора |
явное_предложение_SELECT]

LOOP

 программный код

END LOOP [слово-комментарий];

Объявление индексной_записи в PL/SQL выполняется автоматически с типом имя_курсора%TYPE и самостоятельного объявления не требует.

Goto (безусловный переход)

GOTO имя_метки выражение;

Ограничения области действия GOTO:

- нельзя передавать управление внутрь предложения IF, LOOP и вложенного блока
- нельзя передавать управление из одного раздела предложения IF в другой
- нельзя передавать управление извне/внутрь подпрограммы
- нельзя передавать управление из раздела обработки исключительных состояний в основной раздел блока PL/SQL
- нельзя передавать управление из основного раздела блока PL/SQL в раздел обработки исключительных состояний (это можно делать только с помощью RAISE)

Но передавать управление изнутри предложения IF, LOOP и вложенного блока возможно

Метки в циклах и блоках

```
SQL> BEGIN
2      <<outer>>
3      DECLARE
4          i INTEGER := 1;
5      BEGIN
6          <<inner>>
7          DECLARE
8              i INTEGER := 2;
9          BEGIN
10             dbms_output.put_line('i = '||i); -- 1 или 2?
11             dbms_output.put_line('outer.i = '||outer.i);
12             dbms_output.put_line('inner.i = '||inner.i);
13         END;
14     END;
15 END;
16 /
i = 2
outer.i = 1
inner.i = 2
```

Процедура PL/SQL успешно завершена.