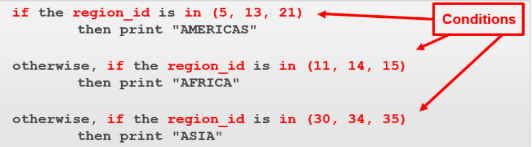
Управляющие конструкции. Операторы IF и CASE, выражение CASE. Использование NULL-значений.

**Оператор IF**

• Оператор IF, показанный ниже, содержит альтернативные варианты действий в блоке по условиям.

• Условие – это выражение с истинным или ложным (TRUE/FALSE) значением, которое используется для принятия решения.

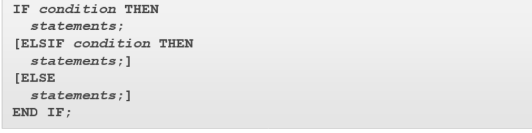


Структура операторов IF

• Структура оператора PL/SQL IF аналогична структуре IF других процессуальных языках.

• Это позволяет PL / SQL выполнять действия выборочно, основываясь на условия.

• Синтаксис:



• Они могут включать в себя дополнительные утверждения, если они содержат несколько вложенных операторы IF, ELSE и ELSIF.

• Операторы в предложении THEN выполняются только в том случае, если условие в связанном предложении IF имеет значение TRUE.

• ELSIF – это ключевое слово, которое вводит дополнительное логическое выражение.

(Если первое условие дает FALSE или NULL, то ключевое слово ELSIF вводит дополнительные условия.

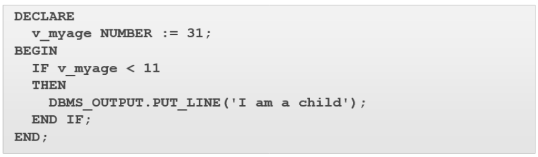
ELSIF – это правильное написание, а не ELSEIF.)

Примечание

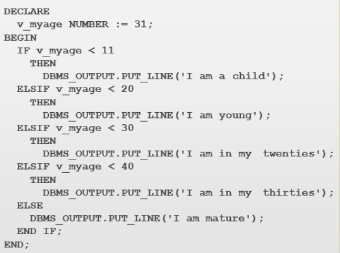
• ELSIF и ELSE являются необязательными в операторе IF. Вы можете иметь любое количество ключевых слов ELSIF, но только одно ELSE в вашем операторе IF.

• END IF обозначает конец оператора IF и должен завершаться точкой с запятой.

Примеры:



Переменная больше 11, что не подходит под условие => ничего не выведется.

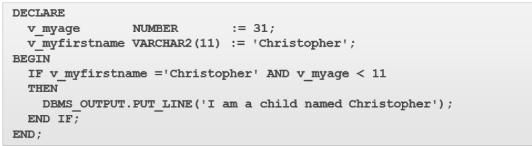


Тут переменная не подходит под первые три условия, но подходит по четвертое, где “< 40”, поэтому выведется ‘I am in my my thirties’.

\* Если у вас есть несколько предложений (как в примере выше) в операторе IF и условие является FALSE или NULL, переходит к следующему предложению.

\* Условия оцениваются один за другим. Если все условия имеют значение FALSE или NULL, то выполняются операторы в предложении ELSE, если такого нет, ничего не выполняется.

Также условия могут быть множественными, используя операторы, такие как: AND, OR или NOT.



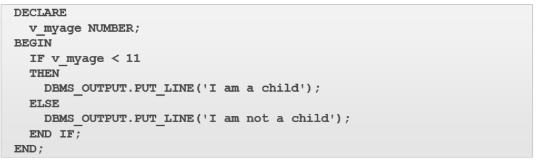
Тут по условию имя у нас совпадает и дает TRUE, но возраст не меньше 11, что дает FALSE: TRUE AND FALSE = FALSE, из чего следует, что ничего не выведется.

**Значение NULL в IF выражениях.**

• В этом примере переменная v myage объявляется, но не инициализируется.

\* Условие в операторе IF возвращает значение NULL, которое не является ни истинным, ни ложным.

• В таком случае элемент управления переходит к оператору ELSE, потому что просто NULL не является истинным.



**Обработка NULLS**

При работе с nulls вы можете избежать некоторых распространенных ошибок, имея в виду следующие правила:

\* Простые сравнения с использованием nulls всегда приводят к NULL.

• Применение логического оператора NOT к null приводит к NULL.

• В операторах условного управления, если условие дает NULL, он ведет себя так же, как и FALSE, и связанная с ним последовательность операторов не выполняется.

Пример:

a := NULL;

b := NULL;

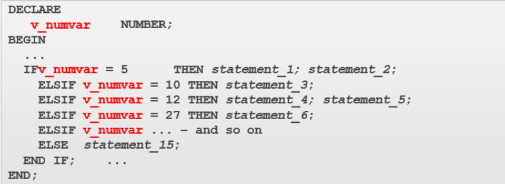
IF a = b THEN …

НЕ ВЫПОЛНИТСЯ, так как сранение двух нуллов равно NULL, а не TRUE.

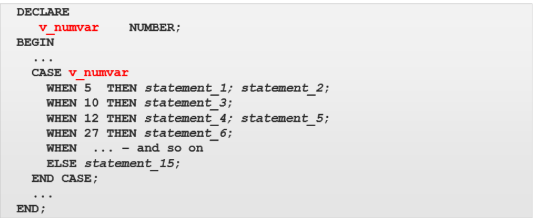
**Оператор CASE.**

Если мы сравниваем единственную переменную с разными условиями, во избежание большого количества повторов мы можем использовать оператор CASE.

Вместо



Это:



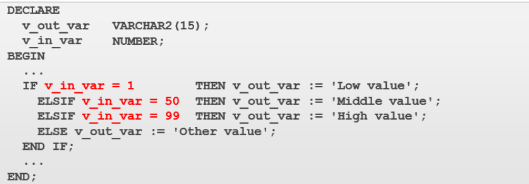
\* Вы можете использовать операторы CASE для проверки условий неравнозначности, таких как <, >, > = и т. д.

\* Синтаксис практически идентичен эквивалентному оператору IF.

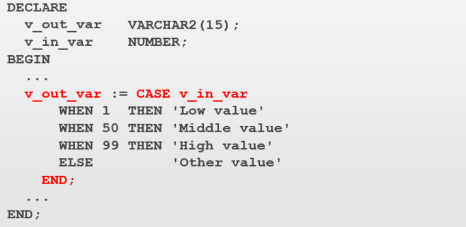
**CASE выражения.**

Когда вы хотите присвоить значение одной переменной, которая зависит от другой, можно использовать CASE выражения. Также как и в случае с CASE оператором.

Вместо



Это:

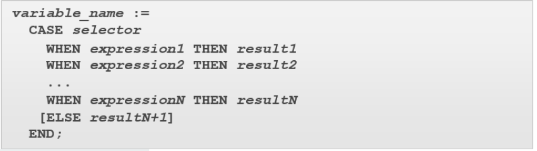


CASE выражение синтаксис:

\* Выражение CASE отличается от оператора CASE тем, что оно выбирает один из нескольких результатов и присваивает его переменной.

• CASE выражение заканчивается END, а не END CASE.

• В синтаксисе expressionN может быть литеральным значением или выражением, например (v other var \* 2).



Чем отличается CASE выражение от CASE оператора?

\* CASE выражения возвращают значение в переменную.

• CASE выражения заканчиваются END;

\* CASE выражения – это один оператор PL / SQL.

\* CASE операторы оценивают условия и выполняют действия.

\* CASE оператор может содержать множество операторов PL / SQL.

• CASE операторы заканчиваются END CASE;.

Логическая таблица:

