**PL/SQL – Programming Constructs**

**Assignment Statements in PL/SQL**

Assignment Statement assigns a value to a variable

***syntax: variable\_name* := *value*;**

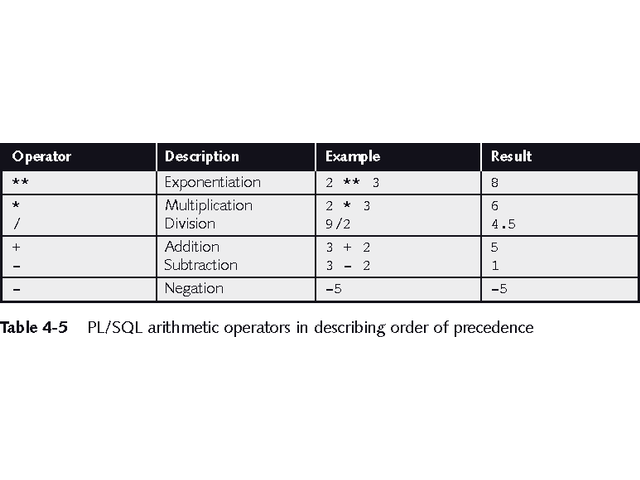
Value can be a literal:

current\_s\_first\_name := 'John';

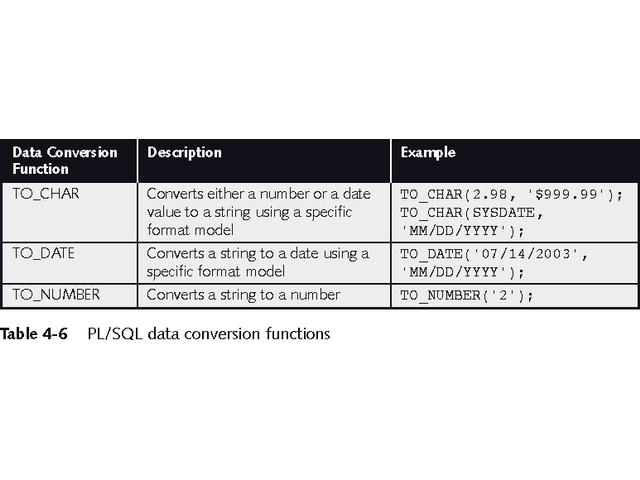
Value can be another variable:

current\_s\_first\_name := s\_first\_name;

**Arithmetic Operators**



**PL/SQL Data Conversion Functions**



**Manipulating Character Strings with PL/SQL**

**To concatenate two strings in PL/SQL, you use the** double bar (||) operator:

*new\_string* := *string1* || *string2*;

To remove blank leading spaces use the LTRIM function:

*string := LTRIM(string\_variable\_name);*

To remove blank trailing spaces use the RTRIM function:

*string := RTRIM(string\_variable\_name);*

To find the number of characters in a character string use the LENGTH function:

*string\_length := LENGTH(string\_variable\_name);*

**To change case, use UPPER, LOWER, INITCAP**

INSTR function searches a string for a specific substring:

*start\_position := INSTR(original\_string, substring);*

SUBSTR function extracts a specific number of characters from a character string, starting at a given point:

*extracted\_string := SUBSTR(string\_variable, starting\_point, number\_of\_characters);*

**Finding and Fixing Logic Errors**

**Locate logic errors by viewing variable values during program execution**

There is no SQL\*Plus debugger

Use DBMS\_OUTPUT statements to print variable values

**PL/SQL Decision Control Structures**

**Use IF/THEN structure to execute code if condition is true**

IF *condition* THEN

*commands that execute if condition is TRUE;*

END IF;

If condition evaluates to NULL it is considered false

Use IF/THEN/ELSE to execute code if condition is true or false

IF *condition* THEN

*commands that execute if condition is TRUE;*

ELSE

*commands that execute if condition is FALSE;*

END IF;

Can be nested – be sure to end nested statements

**Use IF/ELSIF to evaluate many conditions:**

IF *condition1* THEN

*commands that execute if condition1 is TRUE;*

ELSIF *condition2* THEN

*commands that execute if condition2 is TRUE;*

ELSIF *condition3* THEN

*commands that execute if condition3 is TRUE;*

...

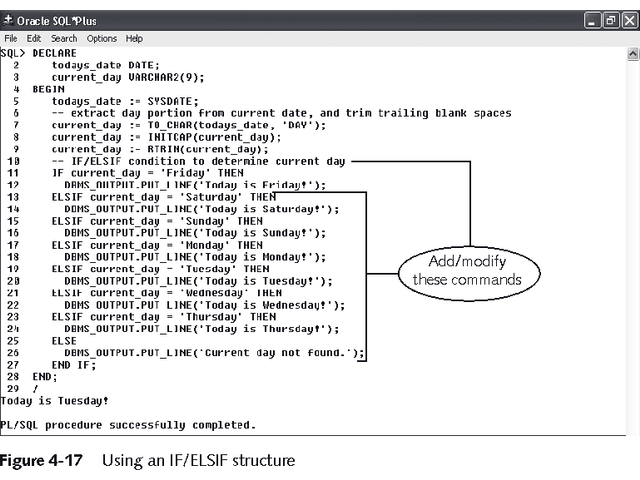
ELSE

*commands that execute if none of the*

*conditions are TRUE;*

END IF;

**IF/ELSIF Example**

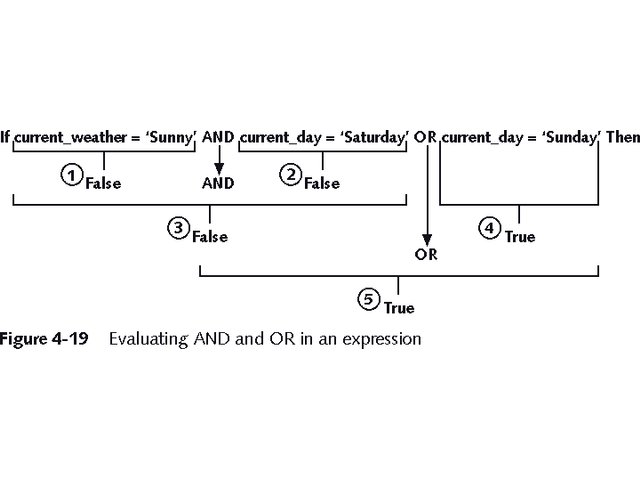


**Complex Conditions**

**Created with logical operators AND, OR and NOT**

AND is evaluated before OR

Use () to set precedence

****

**Using SQL Queries in PL/SQL Programs**

Action queries can be used as in SQL\*Plus

May use variables in action queries

DDL commands may not be used in PL/SQL

**Loops**

Program structure that executes a series of program statements, and periodically evaluates an exit condition to determine if the loop should repeat or exit

**Pretest loop:** evaluates the exit condition before any program commands execute

**Posttest loop:** executes one or more program commands before the loop evaluates the exit condition for the first time

PL/SQL has 5 loop structures

**The LOOP...EXIT Loop**

**LOOP**

[*program statements*]

IF *condition* THEN

EXIT;

END IF;

[*additional program statements*]

**END LOOP**

**The LOOP...EXIT WHEN Loop**

**LOOP**

*program statements*

EXIT WHEN *condition*;

**END LOOP;**

**The WHILE...LOOP**

WHILE *condition* LOOP

*program statements*

END LOOP;

**The Numeric FOR Loop**

FOR *counter\_variable* IN *start\_value* .. *end\_value*

LOOP

*program statements*

END LOOP;

**Programs to be done in Lab**

1. Write a Program to find whether a number is palindrome or not.
2. Write a program to print the table of a given number.
3. Write a program to print Fibonacci series.
4. Write a program to reverse a string.
5. Write a program to find greatest of three numbers.
6. Write a program to display prime numbers from 1 to 50.