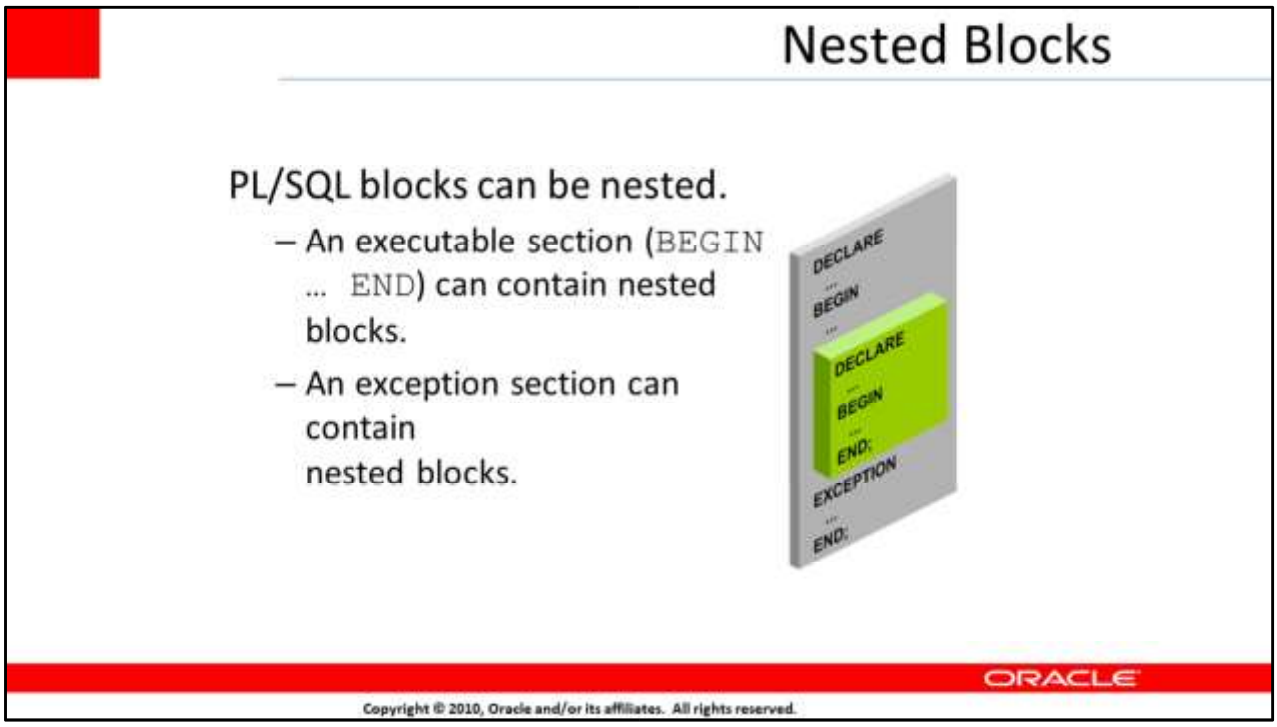
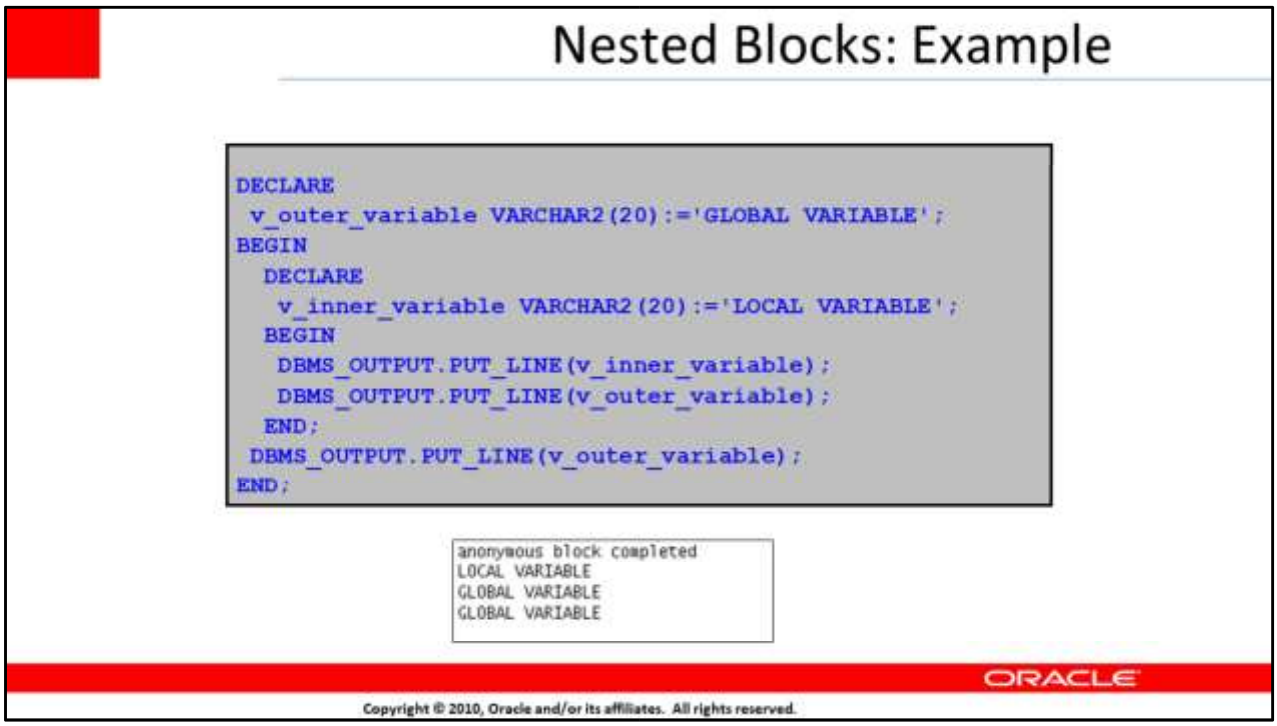


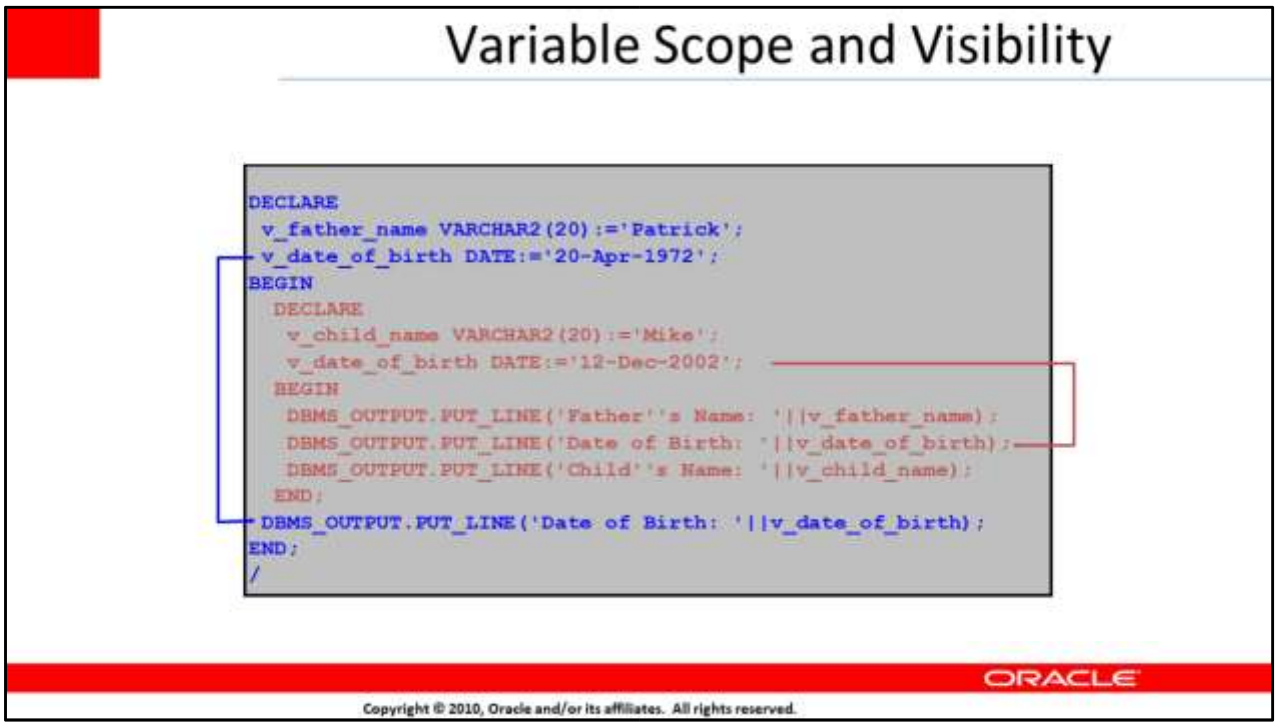
Nested Blocks

Being procedural gives PL/SQL the ability to nest statements. You can nest blocks wherever an executable statement is allowed, thus making the nested block a statement. If your executable section has code for many logically related functionalities to support multiple business requirements, you can divide the executable section into smaller blocks. The exception section can also contain nested blocks.



Nested Blocks (continued)

The example shown in the slide has an outer (parent) block and a nested (child) block. The v_outer_variable variable is declared in the outer block and the v_inner_variable variable is declared in the inner block. v_outer_variable is local to the outer block but global to the inner block. When you access this variable in the inner block, PL/SQL first looks for a local variable in the inner block with that name. There is no variable with the same name in the inner block, so PL/SQL looks for the variable in the outer block. Therefore, v_outer_variable is considered to be the global variable for all the enclosing blocks. You can access this variable in the inner block as shown in the slide. Variables declared in a PL/SQL block are considered local to that block and global to all its subblocks. v_inner_variable is local to the inner block and is not global because the inner block does not have any nested blocks. This variable can be accessed only within the inner block. If PL/SQL does not find the variable declared locally, it looks upward in the declarative section of the parent blocks. PL/SQL does not look downward in the child blocks.



Variable Scope and Visibility

The output of the block shown in the slide is as follows:

anonymous block completed Father's
Name: Patrick
Date of Birth: 12-DEC-02
Child's Name: Mike
Date of Birth: 20-APR-72

Examine the date of birth that is printed for father and child. The output does not provide the correct information, because the scope and visibility of the variables are not applied correctly.

The scope of a variable is the portion of the program in which the variable is declared and is accessible.

The visibility of a variable is the portion of the program where the variable can be accessed without using a qualifier.

Scope

The v_father_name variable and the first occurrence of the v_date_of_birth variable are declared in the outer block. These variables have the scope of the block in which they are declared.

Therefore, the scope of these variables is limited to the outer block.

Variable Scope and Visibility (continued)

Scope (continued)

The v_child_name and v_date_of_birth variables are declared in the inner block or the nested block. These variables are accessible only within the nested block and are not accessible in the outer block. When a variable is out of scope, PL/SQL frees the memory used to store the variable; therefore, these variables cannot be referenced.

Visibility

The v_date_of_birth variable declared in the outer block has scope even in the inner block. However, this variable is not visible in the inner block because the inner block has a local variable with the same name.

1. Examine the code in the executable section of the PL/SQL block. You can print the father's name, the child's name, and the date of birth. Only the child's date of birth can be printed here because the father's date of birth is not visible.
2. The father's date of birth is visible in the outer block and, therefore, can be printed.

Note: You cannot have variables with the same name in a block. However, as shown in this example, you can declare variables with the same name in two different blocks (nested blocks). The two items represented by identifiers are distinct; changes in one do not affect the other.

Using a Qualifier with Nested Blocks

A qualifier is a label given to a block. You can use a qualifier to access the variables that have scope but are not visible.

Example

In the code example:

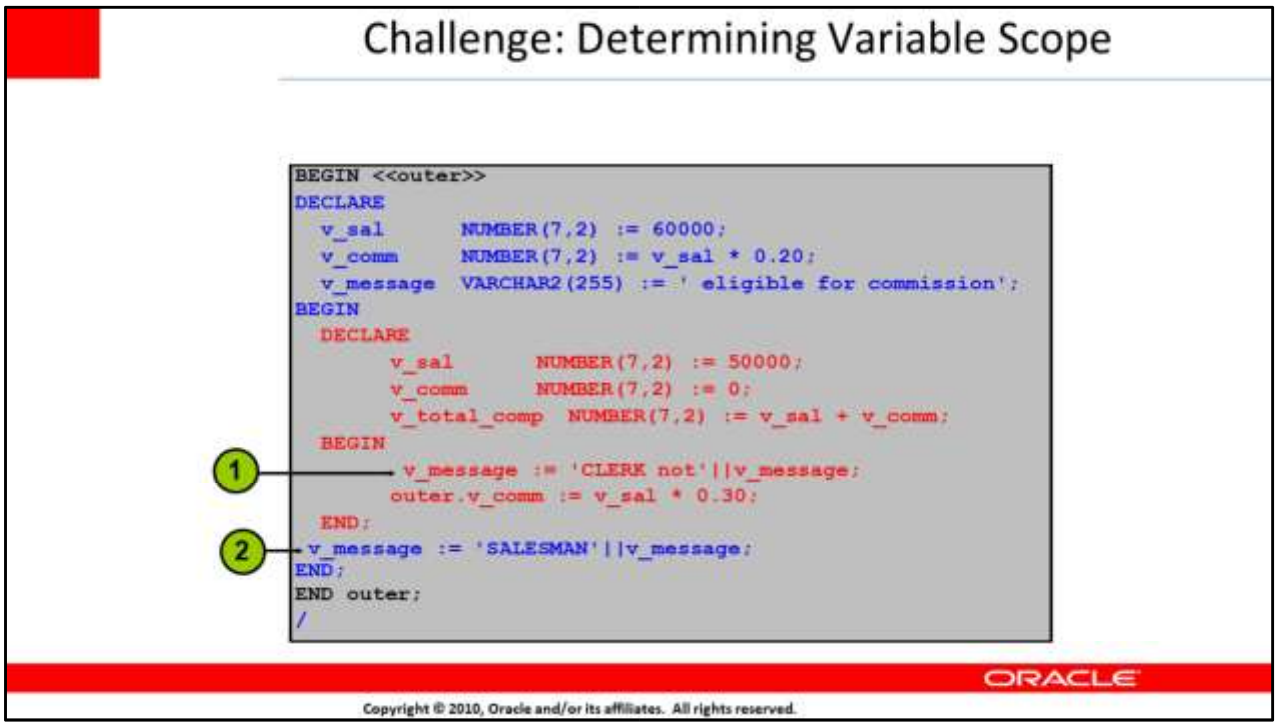
The outer block is labeled outer

Within the inner block, the outer qualifier is used to access the v_date_of_birth variable that is declared in the outer block. Therefore, the father's date of birth and the child's date of birth can both be printed from within the inner block.

The output of the code in the slide shows the correct information:

```
anonymous block completed
Father's Name: Patrick
Date of Birth: 20-APR-72
Child's Name: Mike
Date of Birth: 12-DEC-02
```

Note: La k. You can label any block.



Challenge: Determining Variable Scope

Evaluate the PL/SQL block in the slide. Determine each of the following values according to the rules of scoping:

1. Value of v_message at position 1
2. Value of v_total_comp at position 2
3. Value of v_comm at position 1
4. Value of outer.v_comm at position 1
5. Value of v_comm at position 2
6. Value of v_message at position 2

Answers: Determining Variable Scope

Answers to the questions of scope are as follows:

1. Value of v_message at position 1: CLERK not eligible for commission
2. Value of v_total_comp at position 2: Error. v_total_comp is not visible here because it is defined within the inner block.
3. Value of v_comm at position 1: 0

4. Value of outer.v_comm at position 1: 12000

5. Value of v_comm at position 2: 15000

6. Value of v_message at position 2: SALESMANCLERK not eligible for commission