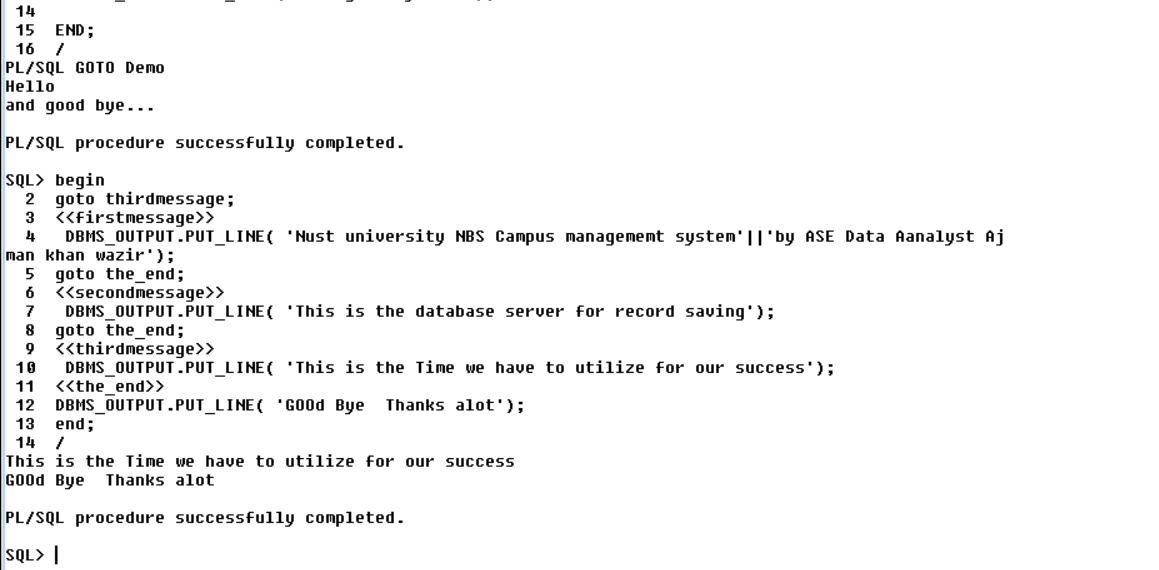
PL/SQL Advanced part 5

First, you cannot use a GOTO statement to transfer control into an [IF](https://www.oracletutorial.com/plsql-tutorial/plsql-if/), [CASE](https://www.oracletutorial.com/plsql-tutorial/plsql-case-statement/) or [LOOP](https://www.oracletutorial.com/plsql-tutorial/plsql-loop/) statement, the same for sub-block.

The following example attempts to transfer control into an IF statement using a GOTO statement:

DECLARE

n\_sales NUMBER;

n\_tax NUMBER;BEGIN

GOTO inside\_if\_statement;

IF n\_sales > 0 THEN

<<inside\_if\_statement>>

n\_tax := n\_sales \* 0.1;

END IF;END;

Code language: SQL (Structured Query Language) (sql)

Oracle issued the following error:

PLS-00375: illegal GOTO statement; this GOTO cannot branch to label 'INSIDE\_IF\_STATEMENT'

Code language: SQL (Structured Query Language) (sql)

Second, you cannot use a GOTO statement to transfer control from one clause to another in the IF statement e.g., from IF clause to ELSIF or ELSE clause, or from one WHEN clause to another in the CASE statement.

The following example attempts to transfer control to a clause in the IF statement:

DECLARE

n\_sales NUMBER;

n\_commission NUMBER;BEGIN

n\_sales := 120000;

IF n\_sales > 100000 THEN

n\_commission := 0.2;

GOTO zero\_commission;

elsif n\_sales > 50000 THEN

n\_commission := 0.15;

elsif n\_sales > 20000 THEN

n\_commission := 0.1;

ELSE

<<zero\_commission>>

n\_commission := 0;

END IF;END;

Code language: SQL (Structured Query Language) (sql)

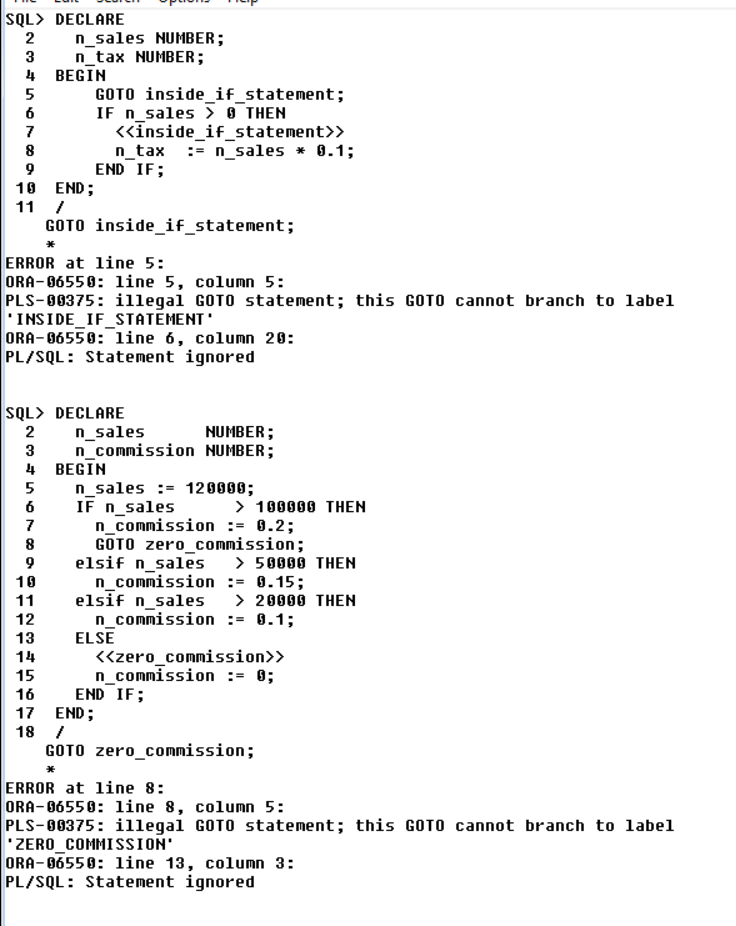
Oracle issued the following error.

PLS-00375: illegal GOTO statement; this GOTO cannot branch to label 'ZERO\_COMMISSION'

Code language: SQL (Structured Query Language) (sql)

Third, you cannot use a GOTO statement to transfer control out of a subprogram or into an [exception handler](https://www.oracletutorial.com/plsql-tutorial/plsql-exception/).

Fourth, you cannot use a GOTO statement to transfer control from an exception handler back into the current block.

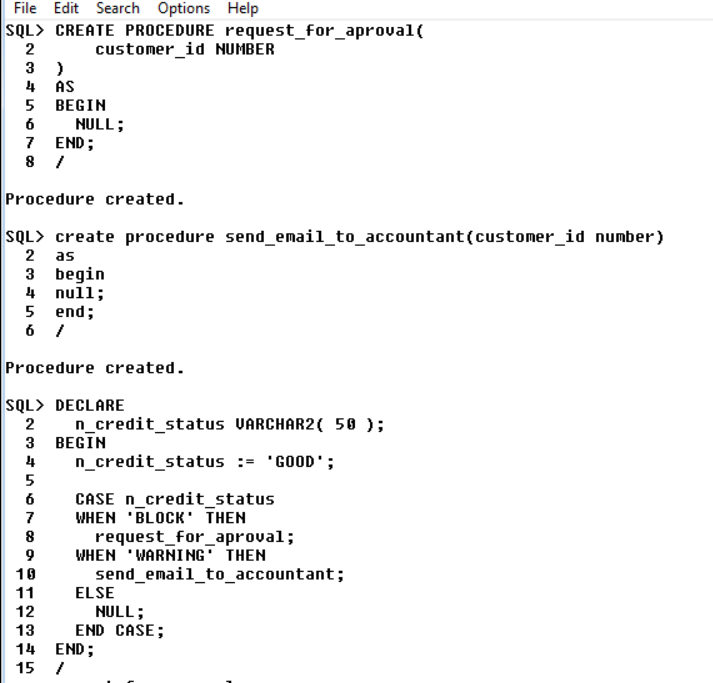
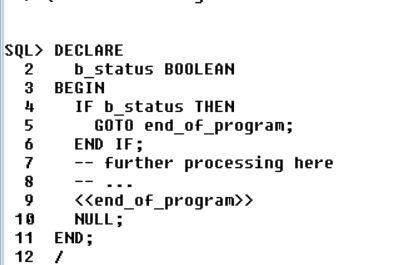
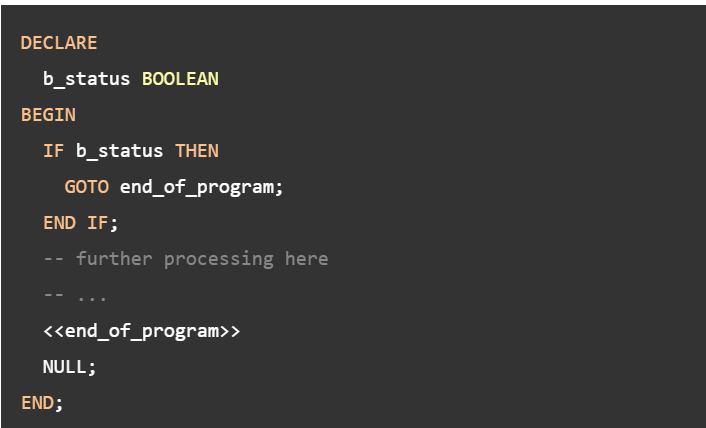
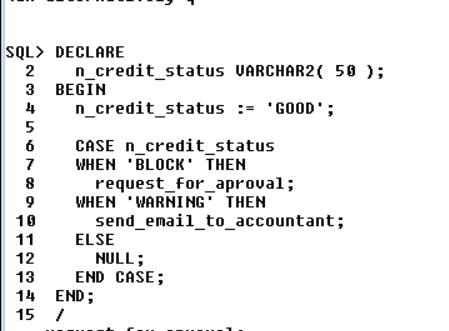
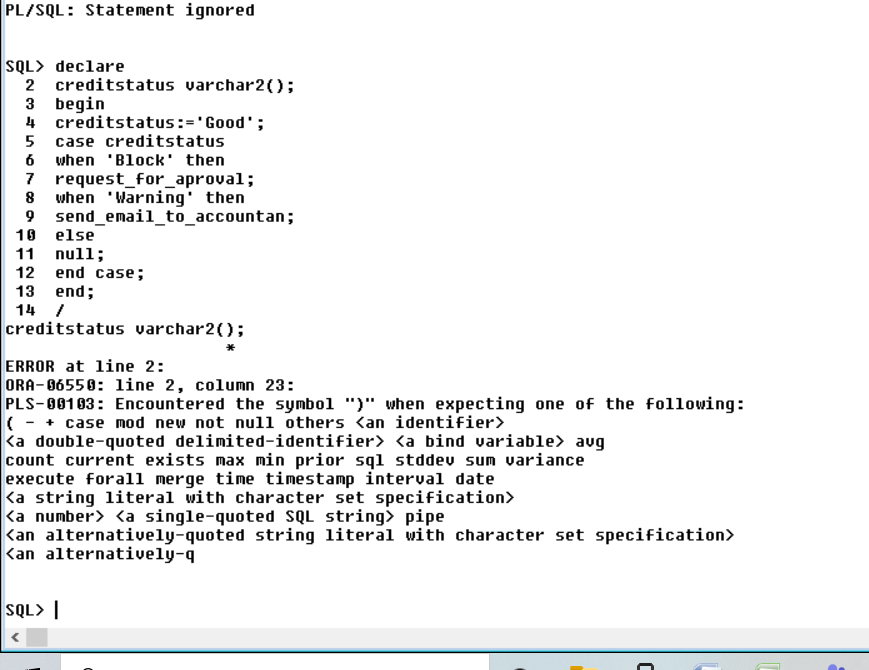
The NULL statement is a NULL keyword followed by a semicolon ( ;). The NULL statement does nothing except that it passes control to the next statement.

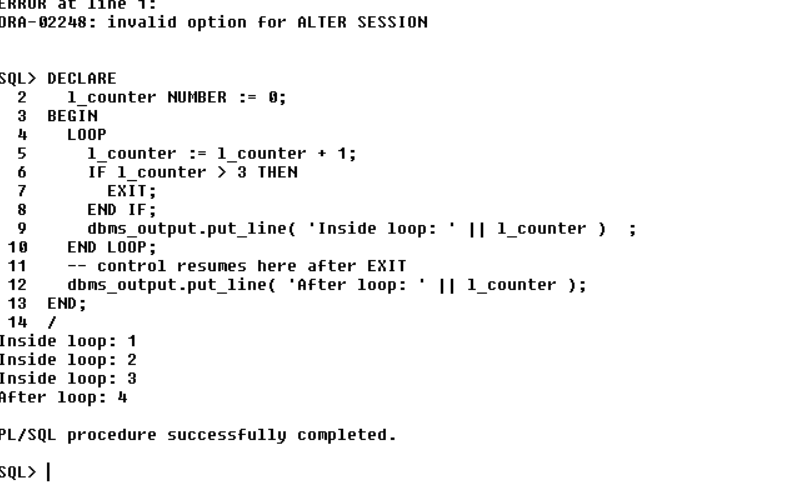
The NULL statement is useful to:

* Improve code readability
* Provide a target for a [GOTO](https://www.oracletutorial.com/plsql-tutorial/plsql-goto/) statement
* Create placeholders for subprograms

## Improving code readability

The following code sends an email to employees whose job titles are Sales Representative



The following explains the logic of the code:

* First, declare and initialize a [variable](https://www.oracletutorial.com/plsql-tutorial/plsql-variables/) l\_counter to zero.
* Second, increase the l\_counter by one inside the loop and exit the loop if the l\_counter is greater than three. If the l\_counter is less than or equal three, show the l\_counter value. Because the initial value of  l\_counter is zero, the code in the body of the loop executes three times before it is terminated.
* Third, display the value of the l\_counter after the loop.

