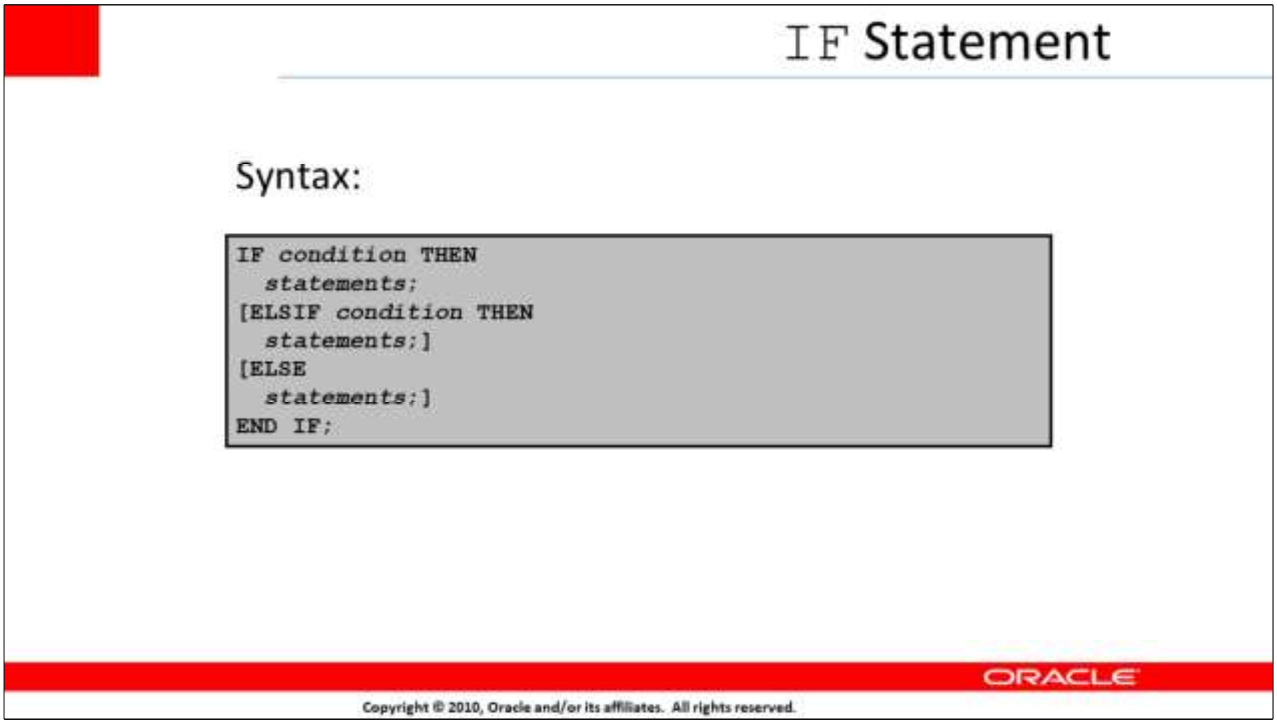


IF Statement

Syntax:

```
IF condition THEN
  statements;
[ELSIF condition THEN
  statements;]
[ELSE
  statements;]
END IF;
```

IF Statement

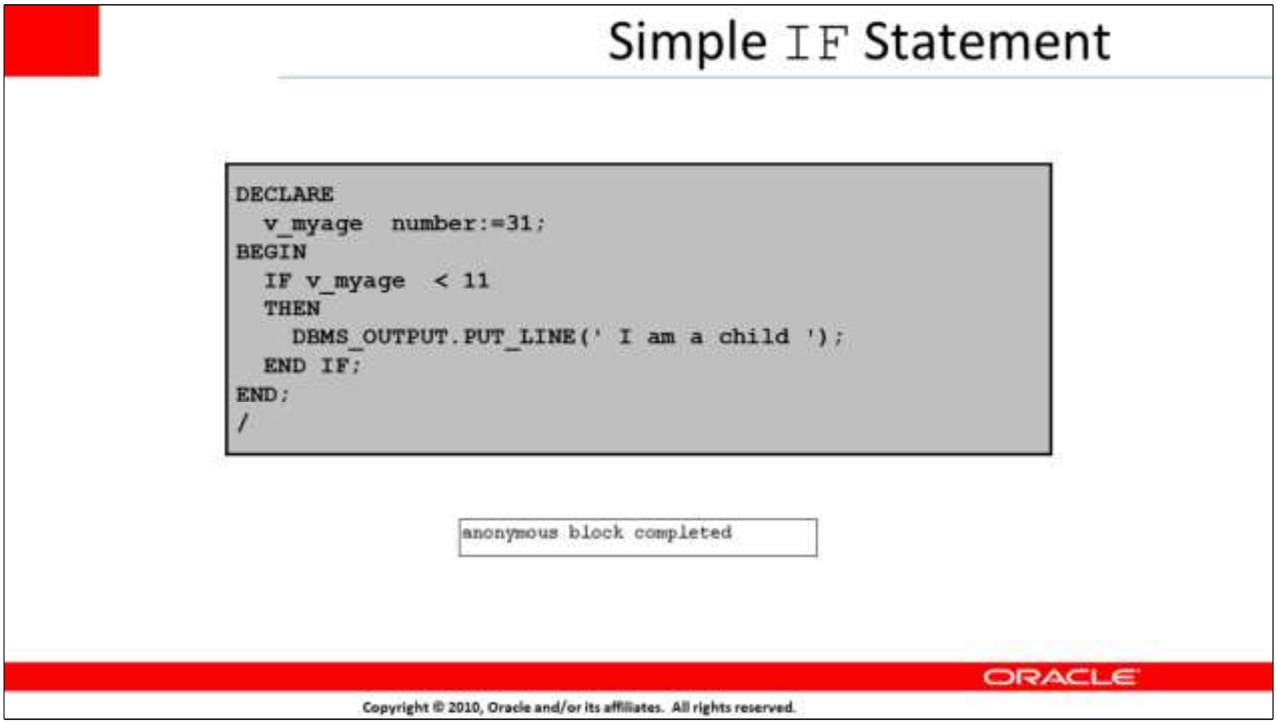
The structure of the PL/SQL IF statement is similar to the structure of IF statements in other procedural languages. It allows PL/SQL to perform actions selectively based on conditions.

In the syntax:

condition — Is a Boolean variable or expression that returns TRUE, FALSE, or NULL

THEN — Introduces a clause that associates the Boolean expression with the sequence of statements that follows it

statements — Can be one or more PL/SQL or SQL statements. (They may include additional IF statements containing several nested IF, ELSE, and ELSIF statements.) The statements in the THEN clause are executed only if the condition in the associated IF clause evaluates to TRUE.



Simple IF Statement

```
DECLARE
  v_myage  number:=31;
BEGIN
  IF v_myage  < 11
  THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  END IF;
END;
/
```

```
anonymous block completed
```

Simple IF Statement

Simple IF Example

The slide shows an example of a simple IF statement with the THEN clause.

The v_myage variable is initialized to 31.

The condition for the IF statement returns FALSE because v_myage is not less than 11.

Therefore, the control never reaches the THEN clause.

Adding Conditional Expressions

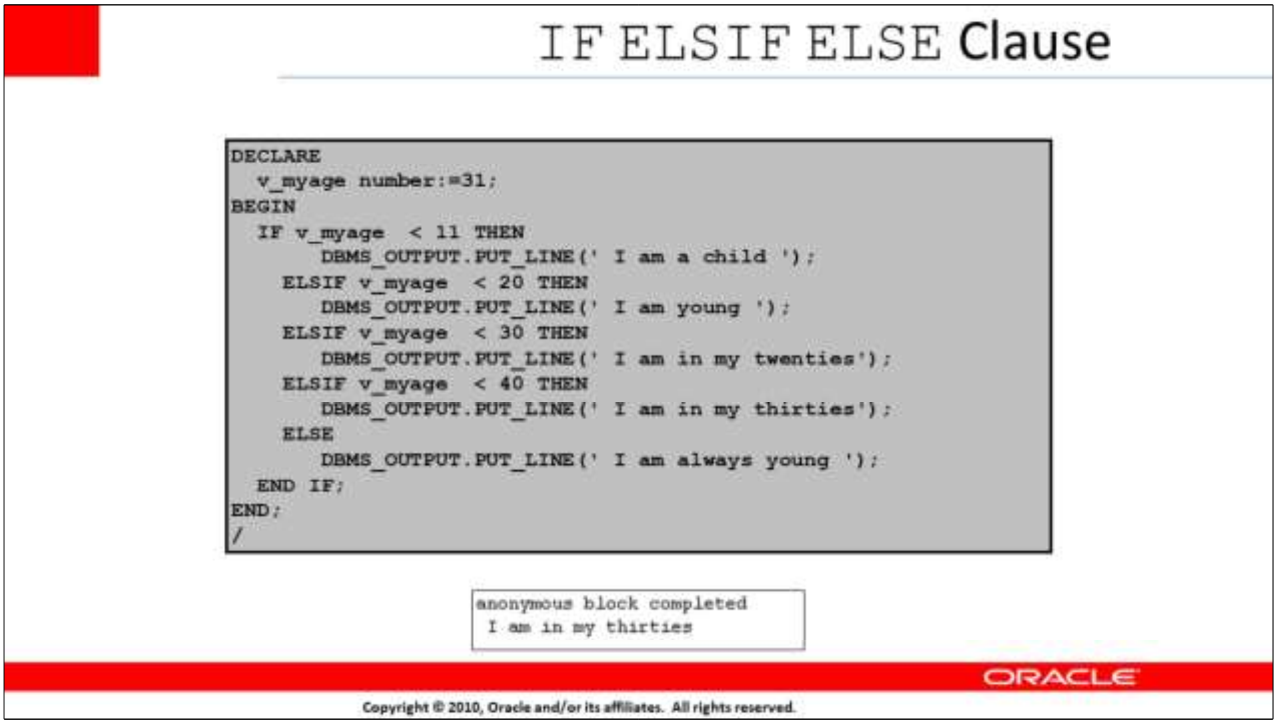
An IF statement can have multiple conditional expressions related with logical operators such as AND, OR, and NOT.

For example:

```
IF (myfirstname='Christopher' AND v_myage <11)
…
```

The condition uses the AND operator and therefore, evaluates to TRUE only if both conditions are evaluated as TRUE. There is no limitation on the number of conditional expressions. However, these statements must be related with appropriate logical operators.

IFTHENELSE Statement



IF ELSIF ELSE Clause

```
DECLARE
  v_myage number:=31;
BEGIN
  IF v_myage  < 11 THEN
       DBMS_OUTPUT.PUT_LINE(' I am a child ');
    ELSIF v_myage  < 20 THEN
       DBMS_OUTPUT.PUT_LINE(' I am young ');
    ELSIF v_myage  < 30 THEN
       DBMS_OUTPUT.PUT_LINE(' I am in my twenties');
    ELSIF v_myage  < 40 THEN
       DBMS_OUTPUT.PUT_LINE(' I am in my thirties');
    ELSE
       DBMS_OUTPUT.PUT_LINE(' I am always young ');
  END IF;
END;
/
```

```
anonymous block completed
 I am in my thirties
```

ORACLE

An ELSE clause is added to the code in the previous slide. The condition has not changed and, therefore, still evaluates to FALSE. Recall that the statements in the THEN clause are executed only if the condition returns TRUE. In this case, the condition returns FALSE and the control moves to the ELSE statement.

The output of the block is shown below the code.

IFELSIFELSE Clause

The IF clause may contain multiple ELSIF clauses and an ELSE clause. The example illustrates the following characteristics of these clauses:

The ELSIF clauses can have conditions, unlike the ELSE clause.

The condition for ELSIF should be followed by the THEN clause, which is executed if the condition for ELSIF returns TRUE. When you have multiple ELSIF clauses, if the first condition is FALSE or NULL, the control shifts to the next ELSIF clause.

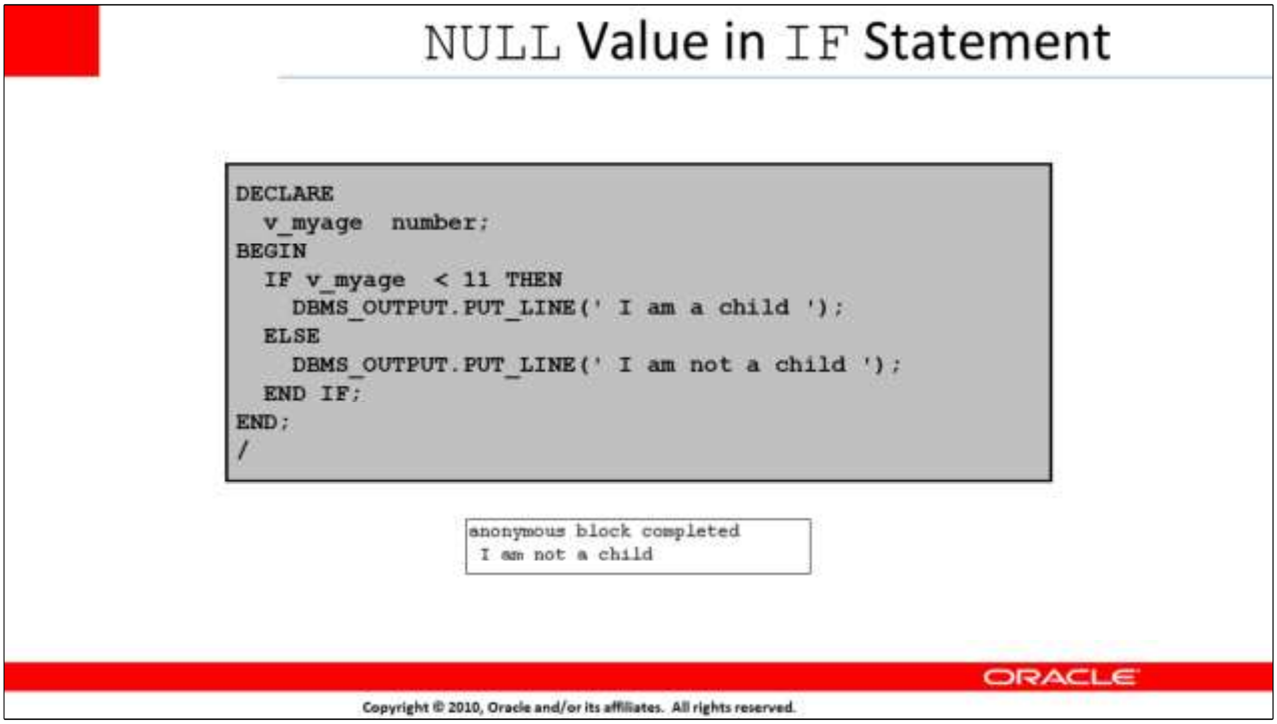
Conditions are evaluated one by one from the top.

If all conditions are FALSE or NULL, the statements in the ELSE clause are executed.

The final ELSE clause is optional.

In the example, the output of the block is shown below the code.

NULL Value in IF Statement



NULL Value in IF Statement

```
DECLARE
  v_myage  number;
BEGIN
  IF v_myage  < 11 THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  ELSE
    DBMS_OUTPUT.PUT_LINE(' I am not a child ');
  END IF;
END;
/
```

```
anonymous block completed
 I am not a child
```

In the example shown in the slide, the variable v_myage is declared but not initialized. The condition in the IF statement returns NULL rather than TRUE or FALSE. In such a case, the control goes to the ELSE statement.

Guidelines

You can perform actions selectively based on conditions that are being met.

When you write code, remember the spelling of the keywords:

–ELSIF is one word.

–ENDIF is two words.

If the controlling Boolean condition is TRUE, the associated sequence of statements is executed; if the controlling Boolean condition is FALSE or NULL, the associated sequence of statements is passed over. Any number of ELSIF clauses is permitted.

Indent the conditionally executed statements for clarity.