

Writing Control Structures

IF Statements

Syntax

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

Simple IF statement:

Set the manager ID to 22 if the employee name is Osborne.

```
IF v_ename = 'OSBORNE' THEN
    v_mgr := 22;
END IF;
```

Simple IF Statements

Set the job title to Salesman, the department number to 35, and the commission to 20% of the current salary if the last name is Miller.

Example

```
...
IF v_ename = 'MILLER' THEN
    v_job := 'SALESMAN';
    v_deptno := 35;
    v_new_comm := sal * 0.20;
END IF;
...
```

IF-THEN-ELSIF Statements

For a given value, calculate a percentage of that value based on a condition.

Example

```
...
IF v_start > 100 THEN
    v_start := 2 * v_start;
ELSIF v_start >= 50 THEN
    v_start := .5 * v_start;
ELSE
    v_start := .1 * v_start;
END IF;
...
```

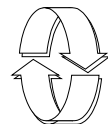
Logic Tables

Build a simple Boolean condition with a comparison operator.

AND	TRUE	FALSE	NULL	OR	TRUE	FALSE	NULL	NOT	
TRUE	TRUE	FALSE	NULL	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	NULL	FALSE	TRUE
NULL	NULL	FALSE	NULL	NULL	TRUE	NULL	NULL	NULL	NULL

Iterative Control: LOOP Statements

- Loops repeat a statement or sequence of statements multiple times.
- There are three loop types:
 - Basic loop
 - FOR loop
 - WHILE loop



Basic Loop

Syntax

```
LOOP                -- delimiter
  statement1;        -- statements
  . . .
  EXIT [WHEN condition]; -- EXIT statement
END LOOP;           -- delimiter
```

where: *condition* is a Boolean variable or expression (TRUE, FALSE, or NULL);

Basic Loop

Example

```
DECLARE
  v_ordid    item.ordid%TYPE := 601;
  v_counter  NUMBER(2) := 1;
BEGIN
  LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_ordid, v_counter);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 10;
  END LOOP;
END;
```

FOR Loop

Syntax

```
FOR counter in [REVERSE]
  lower_bound..upper_bound LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

- Use a FOR loop to shortcut the test for the number of iterations.
- Do not declare the counter; it is declared implicitly.

FOR Loop

Insert the first 10 new line items for order number 601.

Example

```
DECLARE
  v_ordid    item.ordid%TYPE := 601;
BEGIN
  FOR i IN 1..10 LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_ordid, i);
  END LOOP;
END;
```

WHILE Loop

Syntax

```
WHILE condition LOOP ← Condition is
  statement1;           evaluated at the
  statement2;           beginning of
  . . .                 each iteration.
END LOOP;
```

Use the WHILE loop to repeat statements while a condition is TRUE.

WHILE Loop

Example

```
ACCEPT p_new_order PROMPT 'Enter the order number: '
ACCEPT p_items -
  PROMPT 'Enter the number of items in this order: '
DECLARE
  v_count    NUMBER(2) := 1;
BEGIN
  WHILE v_count <= &p_items LOOP
    INSERT INTO item (ordid, itemid)
      VALUES (&p_new_order, v_count);
    v_count := v_count + 1;
  END LOOP;
  COMMIT;
END;
/
```

Nested Loops and Labels

```
...  
BEGIN  
  <<XX>>  
  LOOP  
    v_counter := v_counter+1;  
    EXIT WHEN v_counter>10;  
    <<YY>>  
    LOOP  
      ...  
      EXIT XX WHEN total_done = 'YES';  
      -- Leave both loops  
      EXIT WHEN inner_done = 'YES';  
      -- Leave inner loop only  
      ...  
    END LOOP YY;  
    ...  
  END LOOP XX;  
END;
```

Summary

Change the logical flow of statements by using control structures.

- Conditional (IF statement)
- Loops:
 - Basic loop
 - FOR loop
 - WHILE loop
 - EXIT statement

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.