

若要在PL/SQL中实现控制程序的执行流程和实现复杂的业务逻辑运算，就必须使用流程控制语句，因为只有能够进行结构控制才能灵活实现各种复杂操作和功能。

1 分支结构

1.1 语法

分支结构是最基本的程序结构，分结构由IF语句实现。IF语句有如下的形式：

```
IF 条件1 THEN
    语句序列1;
[ ELSIF 条件2 THEN
    语句序列2;
...
ELSE
    语句序列n; ]
END IF;
```

具体分支结构可以有几下形式：

```
IF-THEN-END IF
IF-THEN-ELSE-END IF
IF-THEN-ELSIF-THEN-ELSE-END IF
```

1.2 示例

1.2.1 IF-THEN-END IF形式

```
DECLARE
    V_SAL NUMBER(7, 2);
BEGIN
    SELECT SAL INTO V_SAL FROM SCOTT.EMP WHERE EMPNO = 7788;
    IF V_SAL >= 3000 THEN
        DBMS_OUTPUT.PUT_LINE('wow rich');
    END IF;
END;
```

1.2.2 IF-THEN-ELSE-END IF形式

```
DECLARE
  V_SEX VARCHAR2(1);
  V_TITIL VARCHAR2(10);
BEGIN
  V_SEX := 'M';
  IF V_SEX = 'm' OR V_SEX = 'M' THEN
    V_TITIL := 'Gentlemen';
  ELSE
    V_TITIL := 'Ladies';
  END IF;
  DBMS_OUTPUT.PUT_LINE('Hello,' || V_TITIL);
END;
```

1.2.3 IF-THEN-ELSIF-THEN-ELSE-END IF形式

```
DECLARE
  V_SAL NUMBER(5);
  V_TAX NUMBER(5, 2);
BEGIN
  SELECT SAL INTO V_SAL FROM SCOTT.EMP WHERE EMPNO = 7788;
  IF V_SAL >= 3000 THEN
    V_TAX := V_SAL * 0.08;
  ELSIF V_SAL >= 1500 THEN
    V_TAX := V_SAL * 0.06;
  ELSE
    V_TAX := V_SAL * 0.04;
  END IF;

  DBMS_OUTPUT.PUT_LINE('Sal:' || V_SAL);
  DBMS_OUTPUT.PUT_LINE('Pay taxes:' || V_TAX);
END;
```

2 选择结构

2.1 值结构

```
DECLARE
  V_JOB VARCHAR2(10);
BEGIN
  SELECT JOB INTO V_JOB FROM SCOTT.EMP WHERE EMPNO = 7788;

  DBMS_OUTPUT.PUT_LINE('Job is' || V_JOB);

  CASE V_JOB
    WHEN 'PRESIDENT' THEN
      DBMS_OUTPUT.PUT_LINE('sss');
    WHEN 'MANAGER' THEN
      DBMS_OUTPUT.PUT_LINE('ss');
    WHEN 'ANALYST' THEN
      DBMS_OUTPUT.PUT_LINE('s');
    ELSE
      DBMS_OUTPUT.PUT_LINE('unknow');
  END CASE;
END;
```

2.2 搜索结构

```
DECLARE
  V_SAL NUMBER(5);
BEGIN
  SELECT SAL INTO V_SAL FROM SCOTT.EMP WHERE EMPNO = 7788;
  DBMS_OUTPUT.PUT_LINE('Sal is ' || V_SAL);

  CASE
    WHEN V_SAL >= 3000 THEN
      DBMS_OUTPUT.PUT_LINE('high');
```

```
WHEN V_SAL >= 1500 THEN
  DBMS_OUTPUT.PUT_LINE('in');
ELSE
  DBMS_OUTPUT.PUT_LINE('low');
END CASE;
END;
```

3 循环结构

3.1 LOOP循环结构

3.1.1 语法

```
LOOP
  语句1;
  语句2;
  ...
EXIT [WHEN 条件]
END LOOP;
```

3.1.2 示例, 求和 $1+2+3+4+\dots+100$

```
DECLARE
  V_TOTAL NUMBER(10):=0;
  V_COUNT NUMBER(5):=1;
BEGIN
  LOOP
    V_TOTAL := V_TOTAL + V_COUNT;

    EXIT WHEN V_COUNT = 100;

    V_COUNT := V_COUNT + 1;

  END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE(V_TOTAL);  
END;
```

3.2 FOR LOOP循环

3.2.1 语法

```
FOR 控制变量 in[REVERSE] 下限..上限  
LOOP  
    语句1;  
    语句2;  
END LOOP;
```

3.2.1 示例

```
DECLARE  
    V_TOTAL NUMBER(10) := 0;  
BEGIN  
    FOR I IN 1 .. 100 LOOP  
        V_TOTAL := V_TOTAL + I;  
    END LOOP;  
  
    DBMS_OUTPUT.PUT_LINE(V_TOTAL);  
END;
```

3.3 WHILE LOOP循环

3.3.1 语法

```
WHILE 条件  
LOOP  
    语句1;  
    语句2;
```

```
...  
END LOOP;
```

3.3.2 示例

```
DECLARE  
  V_COUNT NUMBER(6) := 1;  
BEGIN  
  WHILE V_COUNT < 6 LOOP  
    INSERT INTO SCOTT.EMP (EMPNO, ENAME) VALUES (V_COUNT + 1000, 'oracle');  
    V_COUNT := V_COUNT + 1;  
  END LOOP;  
  COMMIT;  
END;
```