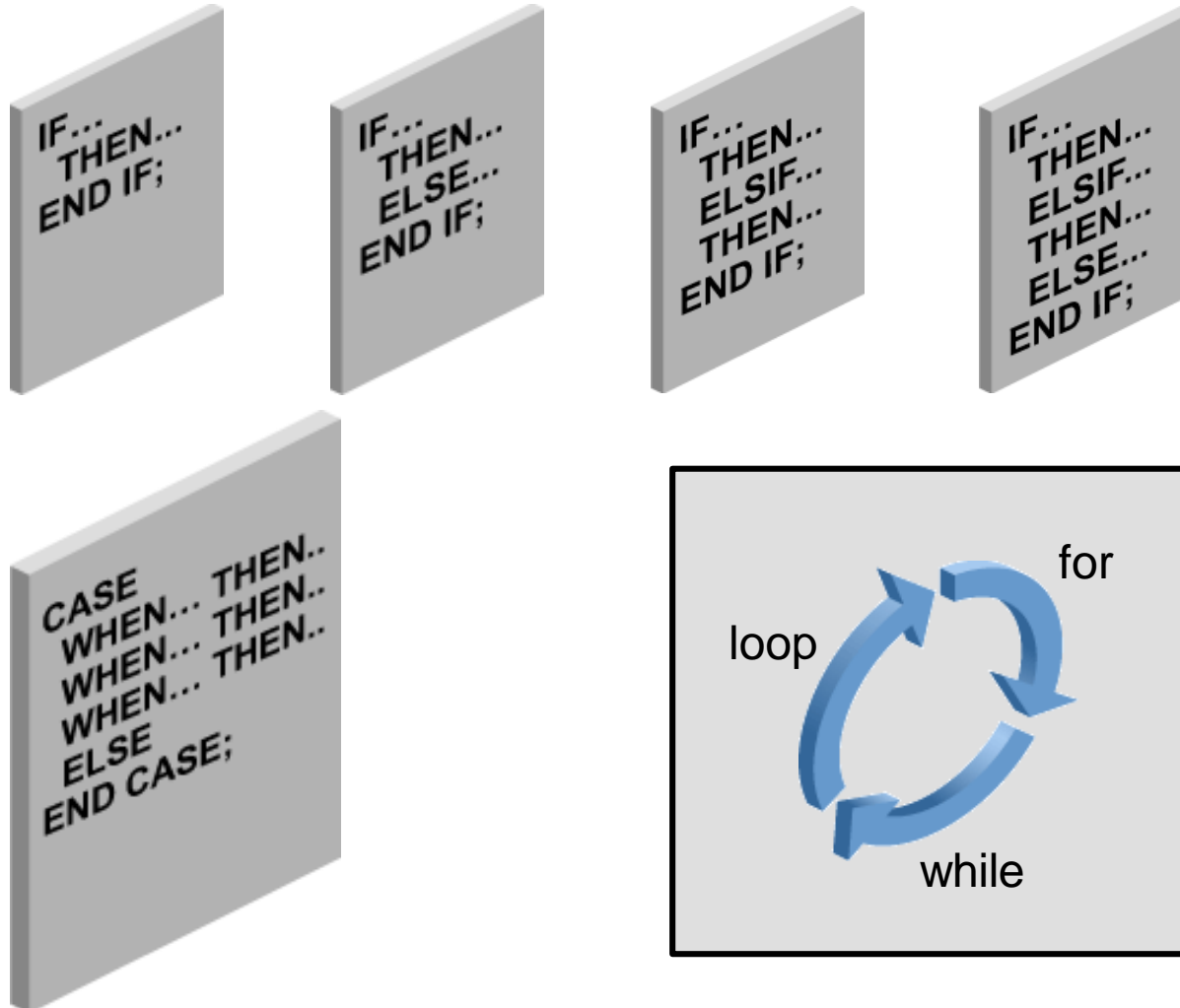


Команды управления исполнением

Управление процессом исполнения



Темы

- Использование выражения IF
- Использование выражения CASE

Выражение IF

Синтаксис:

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

Простой формат выражения IF

```
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
```

Конструкция IF THEN ELSE

```
DECLARE
  v_myage  number:=31;
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
```

Конструкция IF-ELSIF-ELSE

```
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
```

Значение NULL в условии IF

```
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
```


Темы

- Использование выражения IF
- Использование выражения CASE

Выражение CASE

- Выражение CASE позволяет выбрать результат в зависимости от условия.

```
CASE selector  
  WHEN expression1 THEN result1  
  [WHEN expression2 THEN result2  
  ...  
  WHEN expressionN THEN resultN]  
  [ELSE resultN+1]  
END;
```

Выражение CASE: пример

```
DECLARE
    v_grade  CHAR(1) := UPPER('&grade');
    v_appraisal VARCHAR2(20);
BEGIN

    v_appraisal := CASE v_grade
        WHEN 'A' THEN 'Excellent'
        WHEN 'B' THEN 'Very Good'
        WHEN 'C' THEN 'Good'
        ELSE 'No such grade'
    END;
    DBMS_OUTPUT.PUT_LINE ('Grade: ' || v_grade ||
                          'Appraisal' || v_appraisal);
END;
/
```

Выражение CASE с условиями

```
DECLARE
    v_grade CHAR(1) := UPPER('&grade');
    v_appraisal VARCHAR2(20);
BEGIN
    v_appraisal := CASE
        WHEN v_grade = 'A' THEN 'Excellent'
        WHEN v_grade IN ('B','C') THEN 'Good'
        ELSE 'No such grade'

    DBMS_OUTPUT.PUT_LINE ('Grade: ' || v_grade ||
                          ' Appraisal ' || v_appraisal);

END;
/
```

Выражение CASE

```
DECLARE
    v_deptid NUMBER;
    v_deptname VARCHAR2(20);
    v_emps NUMBER;
    v_mngid NUMBER:= 108;
BEGIN
    CASE v_mngid
    WHEN 108 THEN
        SELECT department_id, department_name
            INTO v_deptid, v_deptname FROM departments
            WHERE manager_id=108;
        SELECT count(*) INTO v_emps FROM employees
            WHERE department_id=v_deptid;
    WHEN 200 THEN
        ...
    END CASE;
    DBMS_OUTPUT.PUT_LINE ('You are working in the '|| v_deptname||
    ' department. There are '||v_emps ||' employees in this
    department');
END;
/
```

Обработка Null значений

При работе с NULL значениями можно избежать общих ошибок если соблюдать следующие условия.

- Применение логического оператора `NOT` к `null` возвращает `NULL`.
- Если условие возвращает `NULL`, соответствующая ветка (`CASE`, `IF`) не исполняется.

Таблица логических операторов

Логические выражения можно объединять друг с другом.

AND	<i>TRUE</i>	<i>FALSE</i>	<i>NULL</i>	OR	<i>TRUE</i>	<i>FALSE</i>	<i>NULL</i>	NOT	
<i>TRUE</i>	TRUE	FALSE	NULL	<i>TRUE</i>	TRUE	TRUE	TRUE	<i>TRUE</i>	FALSE
<i>FALSE</i>	FALSE	FALSE	FALSE	<i>FALSE</i>	TRUE	FALSE	NULL	<i>FALSE</i>	TRUE
<i>NULL</i>	NULL	FALSE	NULL	<i>NULL</i>	TRUE	NULL	NULL	<i>NULL</i>	NULL