

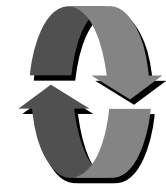
Steuerungsstrukturen

Ablaufsteuerung der PL/SQL Ausführung

Der logische Ablauf der Ausführung kann mit If und Loop Strukturen verändert werden

Bedingte Anweisungen:

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



IF Anweisungen

Syntax

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

Einfaches IF:

Für den Angestellten Osborne soll die Chef ID 22 sein.

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

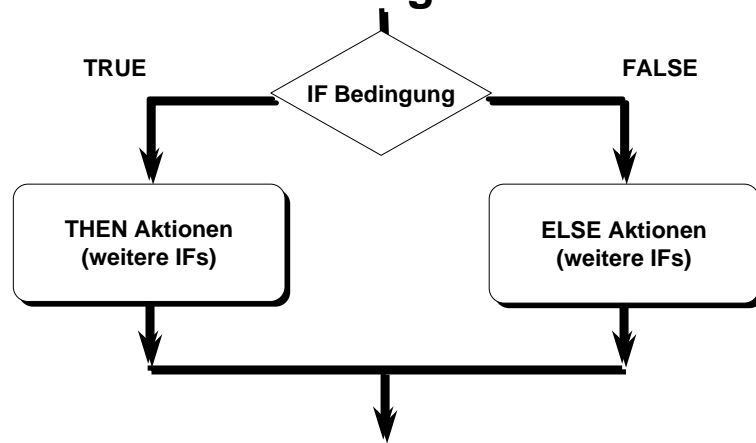
Einfaches IF

Für den Angestellten Miller soll die Tätigkeit Salesman, die Abteilungsnummer 35 und die Provision auf 20% des Gehaltes gesetzt werden.

Beispiel:

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

IF-THEN-ELSE Statement Ausführungsablauf



19-5

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

IF-THEN-ELSE Statements

Setze ein Flag, wenn zwischen Bestell- und Lieferdatum mehr als 5 Tage liegen

Beispiel

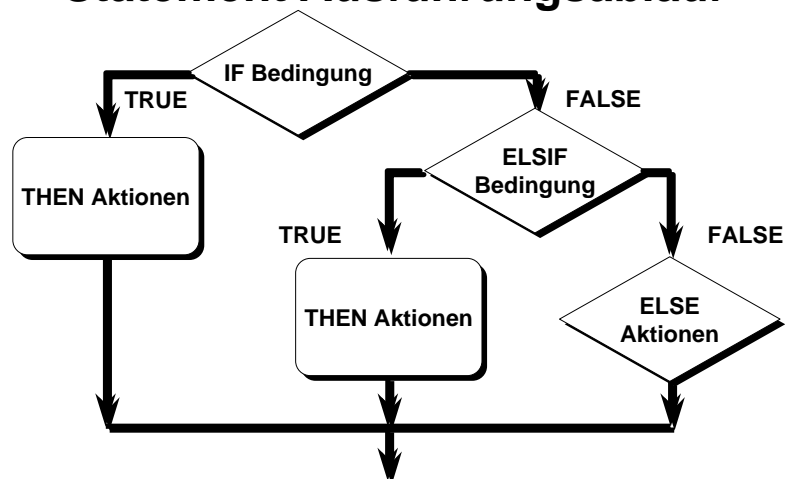
```
...  
IF v_shipdate - v_orderdate < 5 THEN  
    v_ship_flag := 'Acceptable';  
ELSE  
    v_ship_flag := 'Unacceptable';  
END IF;  
...
```

19-6

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

IF-THEN-ELSIF Statement Ausführungsablauf



19-7

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

IF-THEN-ELSIF Statements

Für einen eingegebenen Wert soll ein berechneter Wert zurückgegeben werden.

Beispiel

```
...  
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;  
...
```

19-8

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Logische Bedingungen beschreiben

- Null Werte werden mit dem IS NULL Operator abgefragt.
- Wenn ein Ausdruck einen null-Wert enthält, wird er zu NULL.
- Verkettete Ausdrücke mit null-Werten ergeben einen leeren String.

19-9

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Logik Tabellen

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

19-10

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Boolean Bedingungen

Welchen Wert hat V_FLAG in diesem Falle?

```
v_flag := v_reorder_flag AND v_available_flag;
```

V_REORDER_FLAG	V_AVAILABLE_FLAG	V_FLAG
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
NULL	TRUE	NULL
NULL	FALSE	FALSE

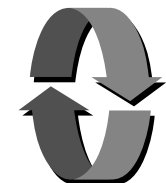
19-11

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Iterative Steuerung: LOOP Statements

- Schleifen wiederholen Anweisungen oder Sequenzen mehrfach.
- Es gibt 3 Schleifentypen:
 - Basis Schleife
 - FOR Schleife
 - WHILE Schleife



19-12

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Basis Schleife

Syntax

```
LOOP                -- Begrenzer
  statement1;      -- Anweisungen
  . . .
  EXIT [WHEN condition]; -- EXIT Anweisung
END LOOP;           -- Begrenzer
```

where: *condition* ist eine Boolean variable oder Ausdruck(TRUE, FALSE, or NULL);

19-13

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

Basis Schleife

Beispiel

```
DECLARE
  v_orcid    item.ordid%TYPE := 101;
  v_counter  NUMBER(2) := 1;
BEGIN
  LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_orcid, v_counter);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 10;
  END LOOP;
END;
```

19-14

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

FOR Schleife

Syntax

```
FOR Zähler in [REVERSE]
  untere_Grenze..obere_Grenze LOOP
  Anweisung1;
  Anweisung2;
  . . .
END LOOP;
```

- FOR-Schleife zur Verkürzung, wenn die Anzahl der Durchläufe bekannt ist.
- Der Index ist implizit deklariert.

19-15

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

FOR Schleife

10 neue Sätze für die Bestellung 101 eintragen

Beispiel

```
DECLARE
  v_orcid    item.ordid%TYPE := 101;
BEGIN
  FOR i IN 1..10 LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_orcid, i);
  END LOOP;
END;
```

19-16

Copyright © Oracle Corporation, 1998. All rights reserved.

ORACLE®

WHILE Schleife

Syntax

```
WHILE bedingung LOOP ← Die Bedingung  
  anweisung1;           wird vor Beginn  
  anweisung2;           des Ablaufs  
  . . .                  überprüft  
END LOOP;
```

Die WHILE-Schleife wiederholt Anweisungen, solange die Bedingung erfüllt ist.

WHILE Schleife

Beispiel

```
ACCEPT p_price PROMPT 'Eingabe des Preises für eine  
Ware: '  
ACCEPT p_itemtot PROMPT 'Eingabe des maximalen  
Gesamtpreises '  
DECLARE  
...  
v_qty          NUMBER(8) := 1;  
v_running_total NUMBER(7,2) := 0;  
BEGIN  
...  
  WHILE v_running_total < &p_itemtot LOOP  
    ...  
    v_qty := v_qty + 1;  
    v_running_total := v_qty * &p_price;  
  END LOOP;  
...  
END;
```

Geschachtelte Schleifen und Marken

- Geschachtelte Schleifen erzeugen mehrere Levels.
- Marken unterscheiden zwischen Blöcken und Schleifen
- Die äußere Schleife wird mit EXIT verlassen

Geschachtelte Schleifen und Marken

```
...  
BEGIN  
  <<Aeussere_Schleife>>      -- Marke  
  LOOP  
    v_counter := v_counter+1;  
    EXIT WHEN v_counter>10;  
    <<Innere_Schleife>>      -- Marke  
    LOOP  
      ...  
      EXIT Aeussere_Schleife WHEN total_done = 'YES';  
      -- verlasse beide Schleifen  
      EXIT WHEN Innere_Schleife = 'YES';  
      -- Verlasse nur die innere Schleife  
      ...  
    END LOOP Innere_Schleife;  
    ...  
  END LOOP Aeussere_Schleife;  
END;
```

Zusammenfassung

Änderung des logischen Ablaufs durch Steuerstrukturen

- **Bedingte Anweisung (IF statement)**
- **Schleifen**
 - **Basis Schleife**
 - **FOR Schleife**
 - **WHILE Schleife**
 - **EXIT Anweisung**