

СУЧАСНІ СУБД

Лекція №11

Тема:

ORACLE PL/SQL

Структура програми

Керування послідовністю виконання програмного коду

Команди умовного розгалуження

До команди, що керують послідовністю виконання програмного коду в залежності від заданих умов, відносяться команди **IF-THEN-ELSE** і **CASE**. Також існують так звані **CASE**-вирази, які іноді дозволяють обійтися без команд **IF** і **CASE**.

Команди безумовного розгалуження

Команда безумовного переходу: **GOTO** і команда холостого ходу **NULL** (не виконує ніяких дій).

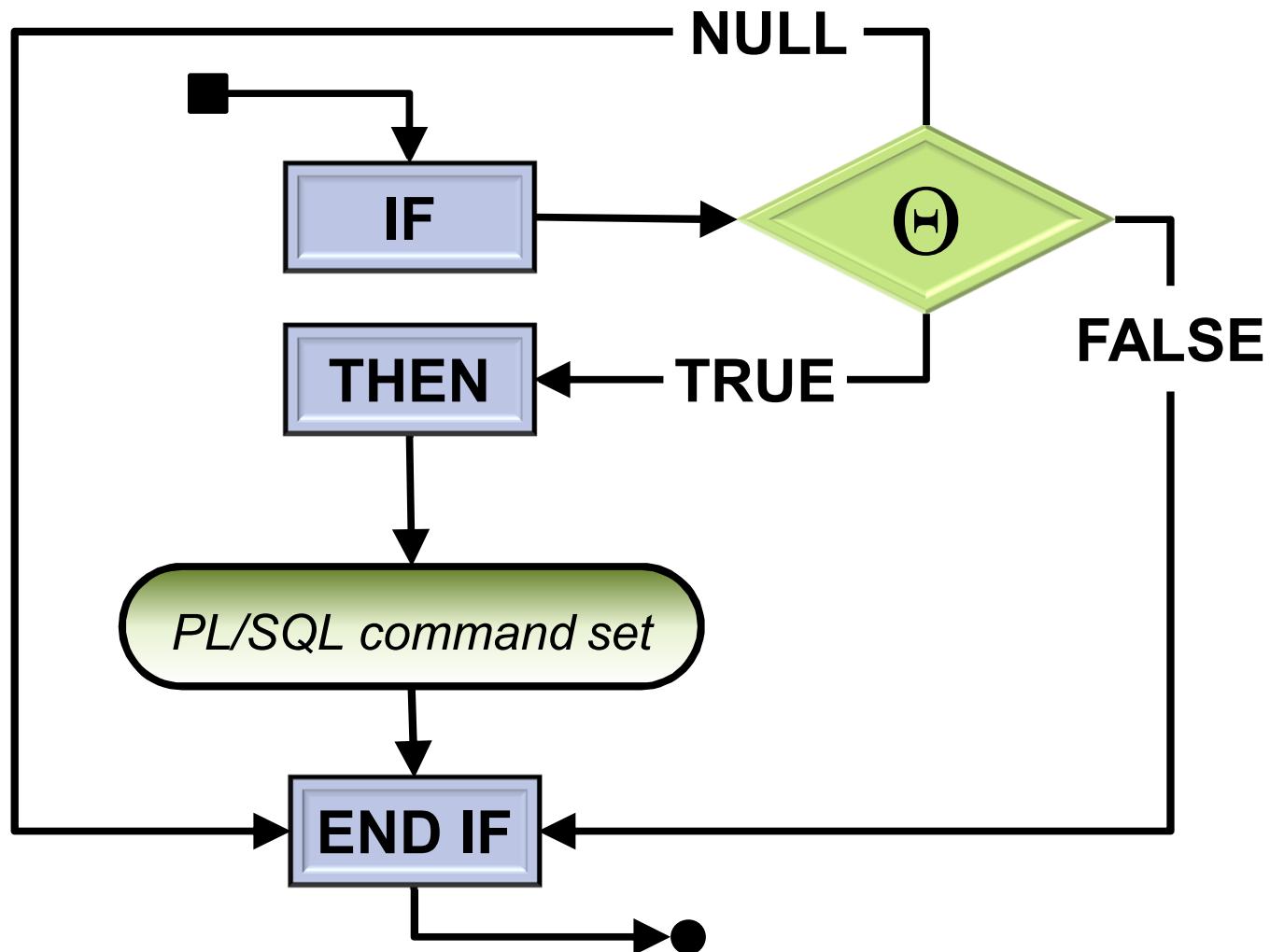
Команди ітеративного розгалуження

Структури що керують багаторазовим (ітеративним) виконання одного і того ж фрагмента програмного коду, називаються циклами. PL/SQL підтримує цикли трьох видів: прості (нескінченні), **FOR** і **WHILE**. Також в Oracle11g з'явилася команда **CONTINUE**.

Команди IF

№	Різновид IF	Характеристики
1.	IF THEN END IF;	<i>Найпростіша форма команди IF. Умова між IF та THEN визначає, чи повинна виконуватися група команд, що знаходитьться між THEN і END IF. Якщо результат перевірки умови дорівнює FALSE або NULL, то код не виконується</i>
2.	IF THEN ELSE END IF;	<i>Реалізація логіки «або-або». Залежно від умови між ключовими словами IF і THEN виконується або код, що знаходитьться між THEN і ELSE, або код між ELSE і END IF. У будь-якому випадку виконується тільки одна з двох груп виконуваних команд.</i>
3.	IF THEN ELSIF ELSE END IF;	<i>Остання, і найскладніша, форма IF вибирає дію з набору взаємовиключних умов і виконує відповідну групу виконуваних команд. У версії Oracle9i і вище її доцільніше замінити командою вибору CASE</i>

Комбінація IF-THEN



Комбінація IF-THEN. Приклад

IF salary > 40000 THEN give_bonus (employee_id,500); END IF;

IF

salary > 40000

THEN

INSERT INTO employee_bonus

 (

employee_id,

bonus_amt

)

VALUES (employee_id, 500);

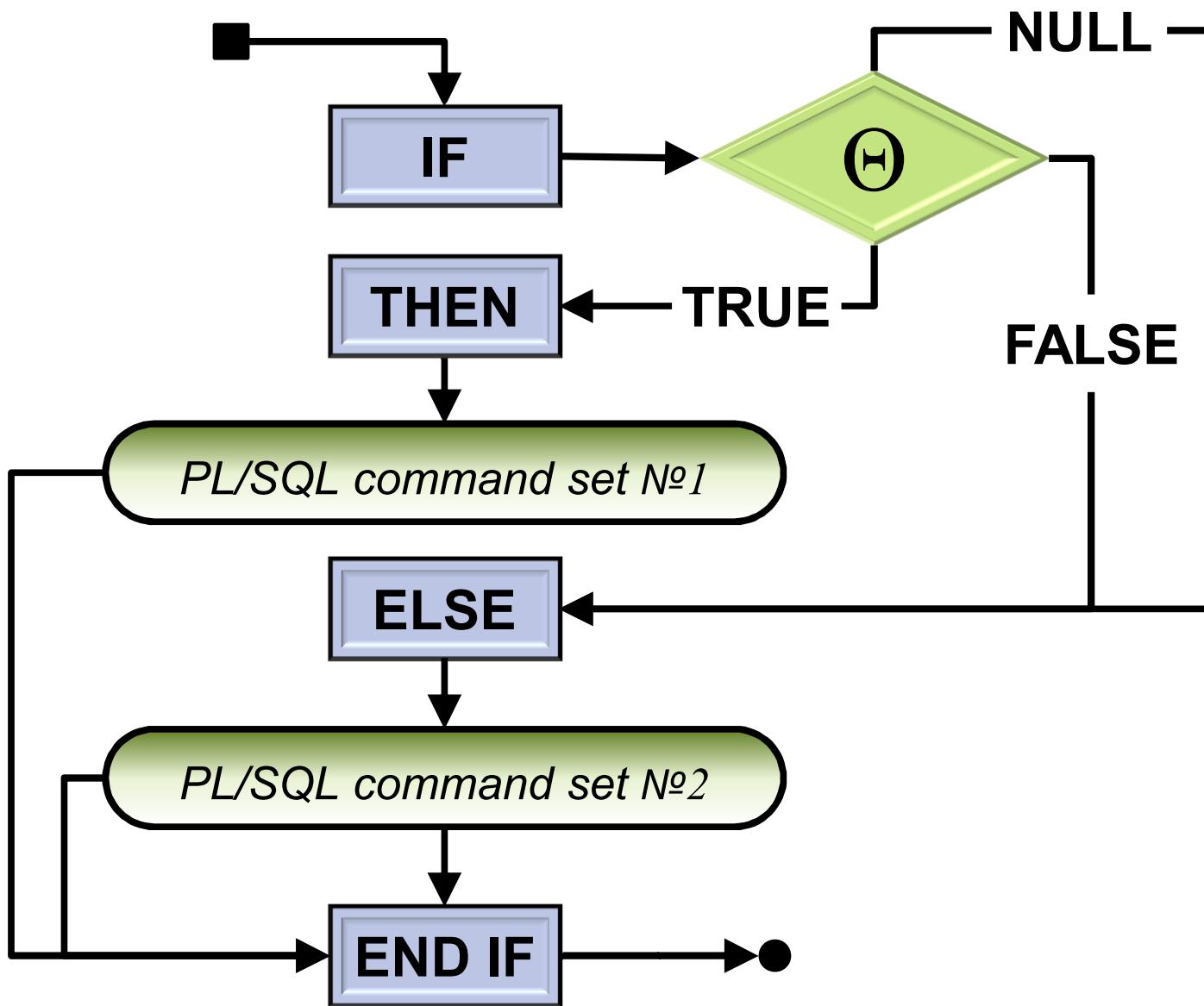
UPDATE emp_employee

SET emp_bonus_given=1

WHERE emp_employee_id=employee_id;

END IF;

Комбінація IF-THEN-ELSE



Комбінація IF-THEN-ELSE. Приклад.

```
IF  
    salary <= 40000  
THEN  
    give_bonus (employee_id, 0);  
ELSE  
    give_bonus (employee_id, 500);  
END IF;
```

```
IF  
    NVL(salary,0) <= 40000  
THEN  
    give_bonus (employee_id, 0);  
ELSE  
    give_bonus (employee_id, 500);  
END IF;
```

Логічні семафори.

IF

:customer.order_total > max_allowable_order

THEN

order_exceeds_balance := TRUE;

ELSE

order_exceeds_balance := FALSE;

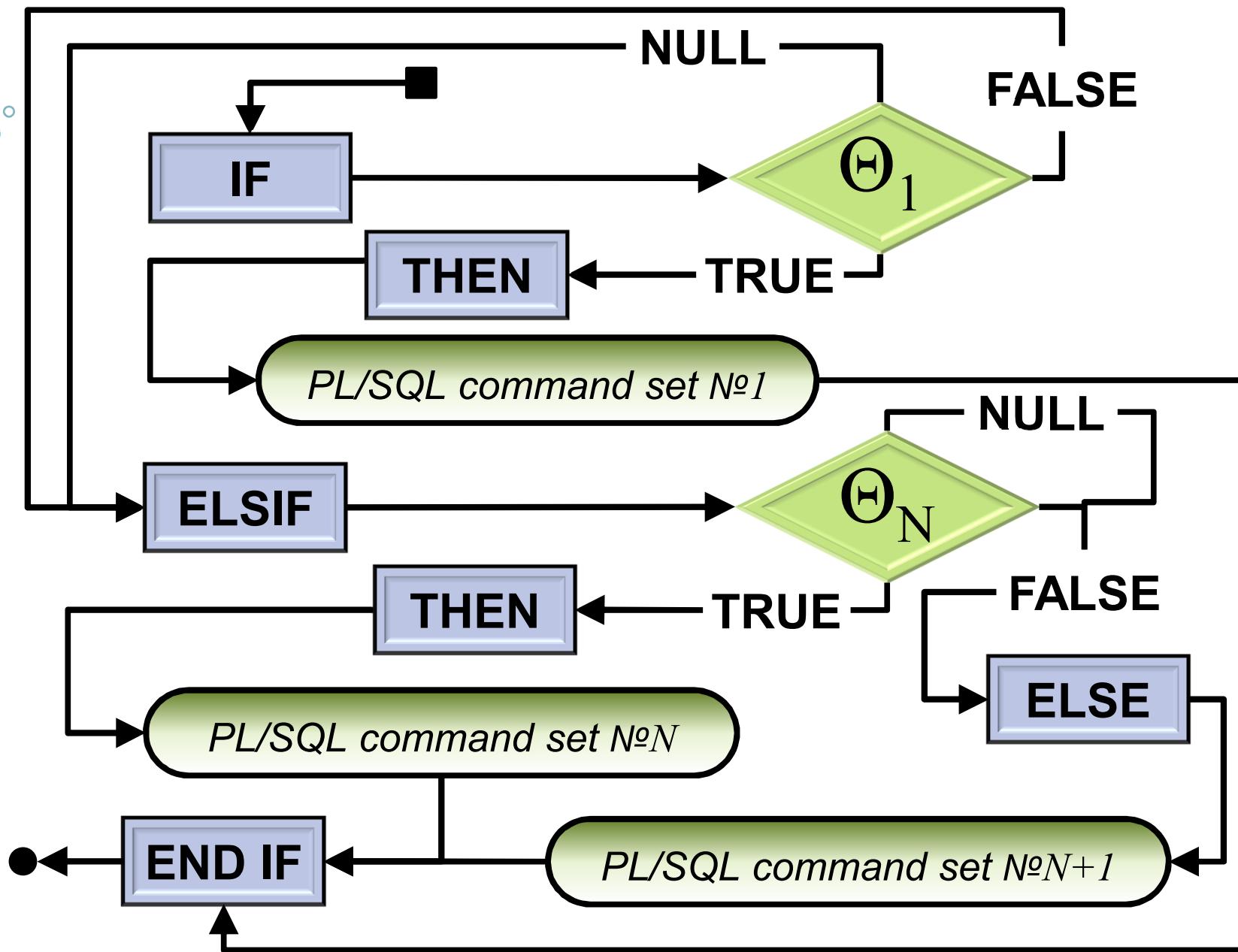
END IF;

order_exceeds_balance :=

(:customer.order_total > max_allowable_order);

IF order_exceeds_balance **THEN** ... **END IF;**

Комбінація IF-THEN-ELSIF-[ELSE]



Комбінація IF-THEN-ELSIF. Приклади

```

IF      salary BETWEEN 10000 AND 20000
THEN    give_bonus(employee_id, 1500);
ELSIF  salary BETWEEN 20000 AND 40000
THEN    give_bonus(employee_id, 1000);
ELSIF  salary > 40000
THEN    give_bonus(employee_id, 500);
ELSE    give_bonus(employee_id, 0);
END IF;

```

```

IF      salary  $\geq 10000$  AND salary  $\leq 20000$ 
THEN    give_bonus(employee_id, 1500);
ELSIF  salary > 20000 AND salary  $\leq 40000$ 
THEN    give_bonus(employee_id, 1000);
ELSIF  salary > 40000
THEN    give_bonus(employee_id, 400);
ELSE    give_bonus(employee_id, 0);
END IF;

```

Вкладені команди IF. Приклади

IF condition №1

THEN

IF condition №2

THEN

command set №2

ELSE

IF condition №3

THEN

command set №3

ELSIF condition №4

THEN

command set №

END IF;

END IF;

END IF;

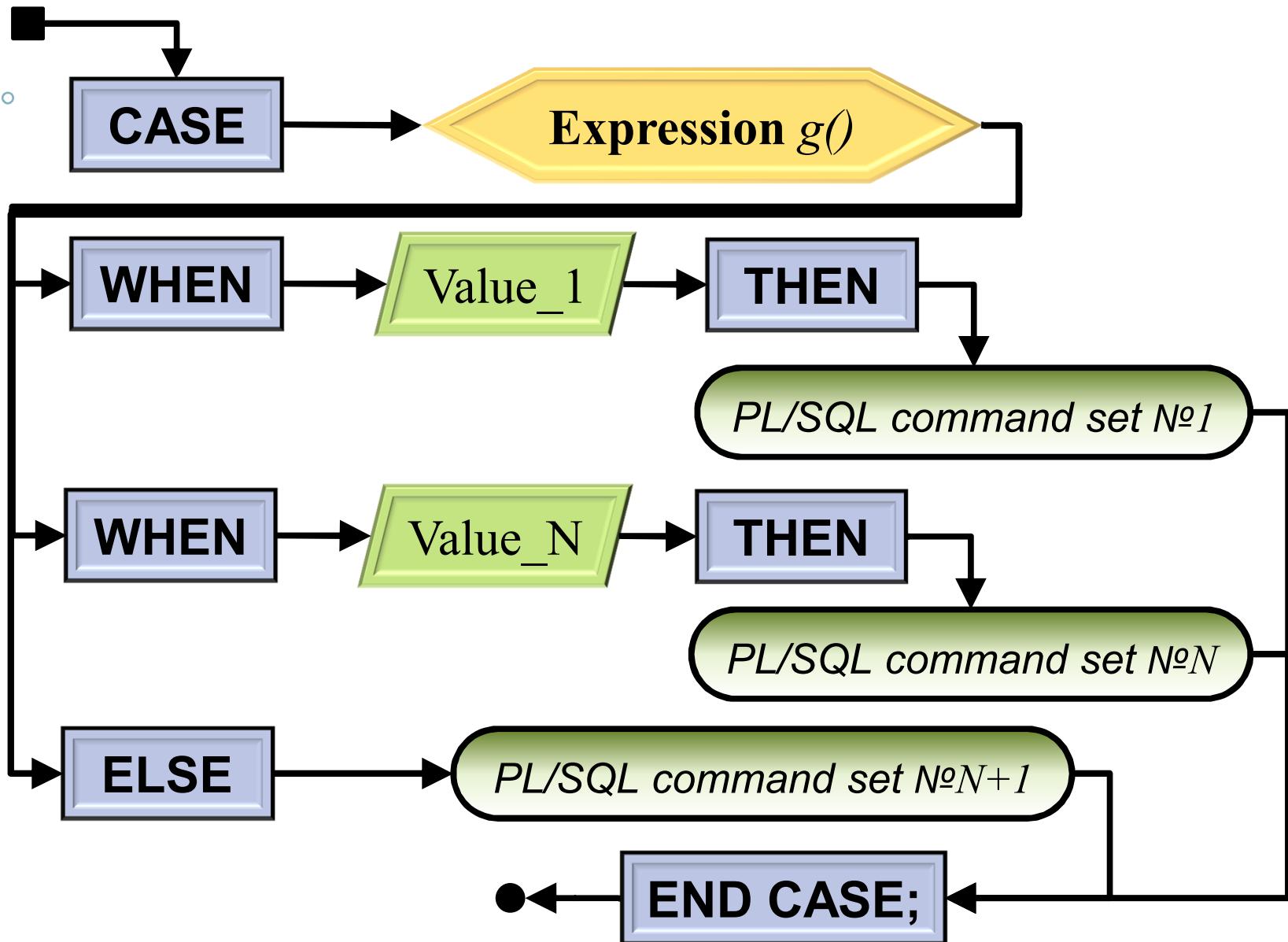
Команди IF. Пришвидшене обчислення

```
IF  
    condition №1  
    AND  
    condition №2  
    AND  
    condition №3  
    .....  
    AND  
    condition №N  
THEN  
    command set №1  
ELSE  
    command set №2  
END IF;
```

Команди і вирази CASE

№	Різновид CASE	Характеристики
1.	Простий CASE;	<i>Пов'язує одну або декілька послідовностей команд PL/SQL з відповідними значеннями (виконувана послідовність вибирається з урахуванням результату обчислення виразу, що повертає одне ззначення).</i>
2.	Пошуковий CASE;	<i>Вибирає для виконання одну або декілька послідовностей команд в залежності від результатів перевірки списку логічних значень. Виконується послідовність команд, пов'язана з першою умовою, результат перевірки якого виявився рівним TRUE</i>

Проста команда CASE. Синтаксис.



Проста команда CASE. Приклад.

```
CASE employee_type
WHEN 'S' THEN
    award_salary_bonus(employee_id);
WHEN 'H' THEN
    award_hourly_bonus(employee_id);
WHEN 'C' THEN
    award_commissioned_bonus(employee_id);
ELSE
    RAISE invalid_employee_type;
END CASE;
```

Проста команда CASE. Приклад.

CASE TRUE

WHEN salary ≥ 10000 **AND** salary ≤ 20000

THEN

give_bonus(employee_id, 1500);

WHEN salary > 20000 **AND** salary ≤ 40000

THEN

give_bonus(employee_id, 1000);

WHEN salary > 40000

THEN

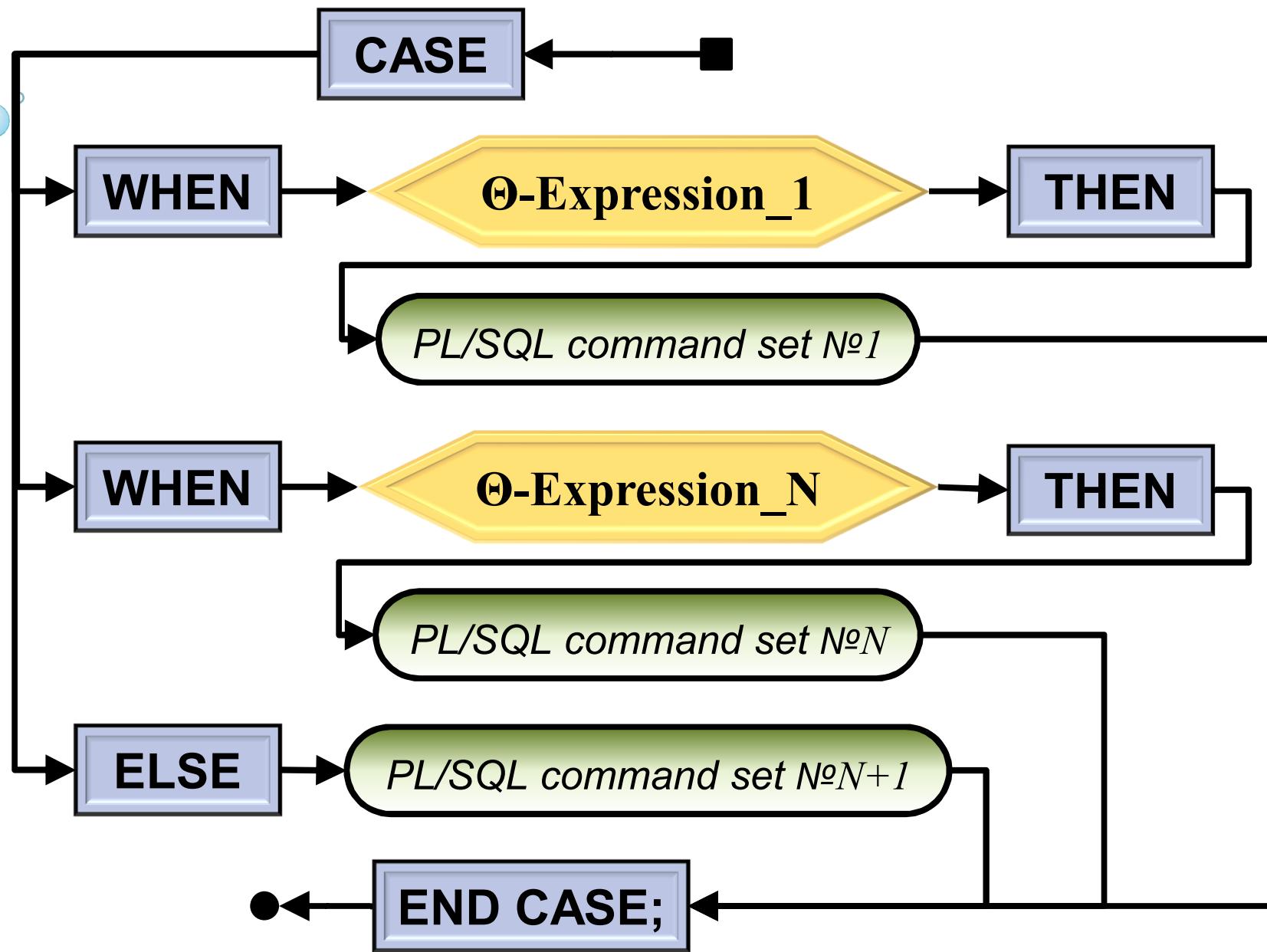
give_bonus(employee_id, 500);

ELSE

give_bonus(employee_id, 0);

END CASE;

Пошукова команда CASE. Синтаксис.



Пошукова команда CASE. Приклад.

CASE

WHEN salary >= 10000 **AND** salary <=20000

THEN

give_bonus(employee_id, 1500);

WHEN salary > 20000 **AND** salary <= 40000

THEN

give_bonus(employee_id, 1000);

WHEN salary > 40000

THEN

give_bonus(employee_id, 500);

ELSE

give_bonus(employee_id, 0);

END CASE;

Пошукова команда CASE. Приклад.

CASE

WHEN salary > 40000 **THEN**

give_bonus(employee_id, 500);

WHEN salary > 20000 **THEN**

give_bonus(employee_id, 1000);

WHEN salary >= 10000 **THEN**

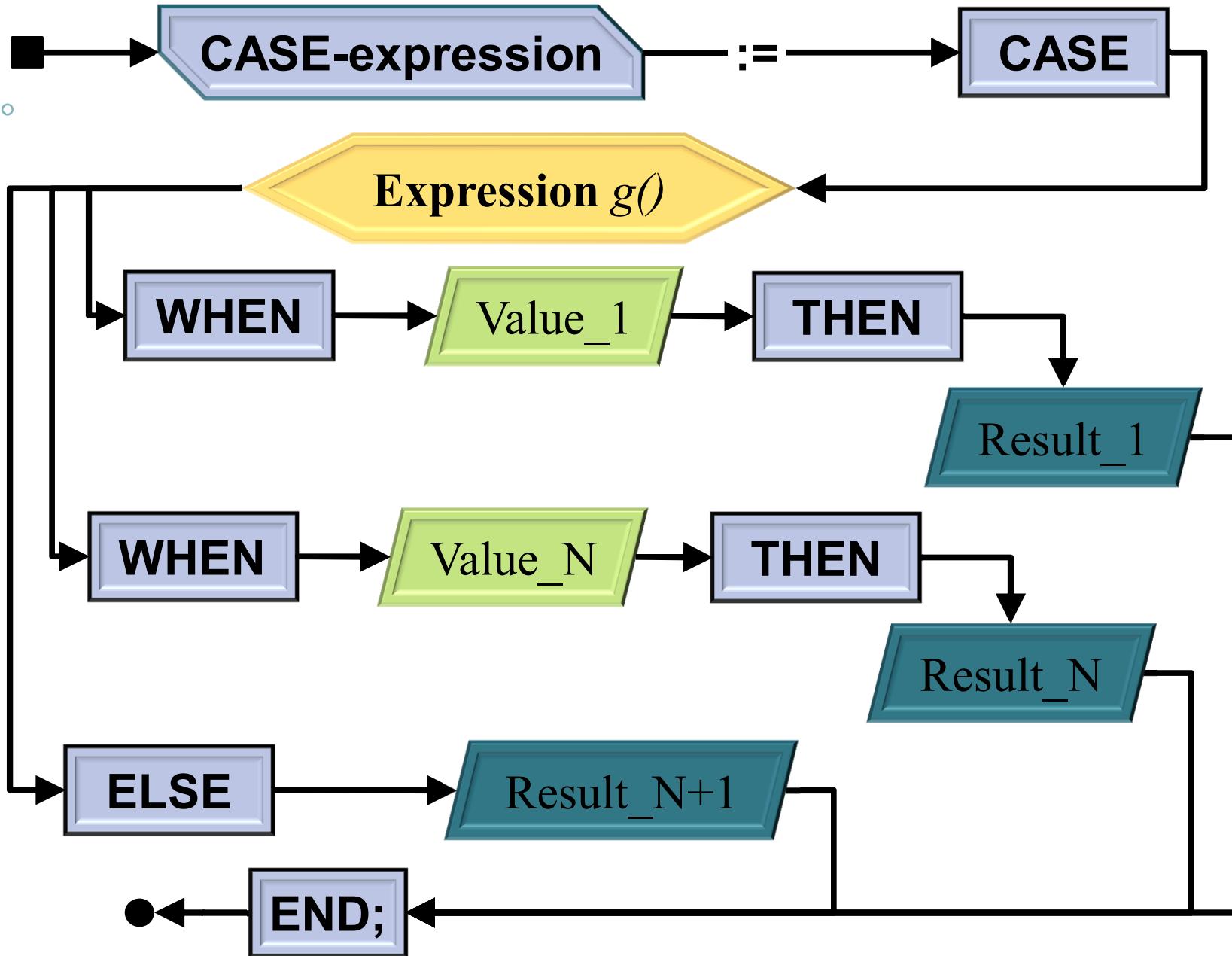
give_bonus(employee_id, 1500);

ELSE

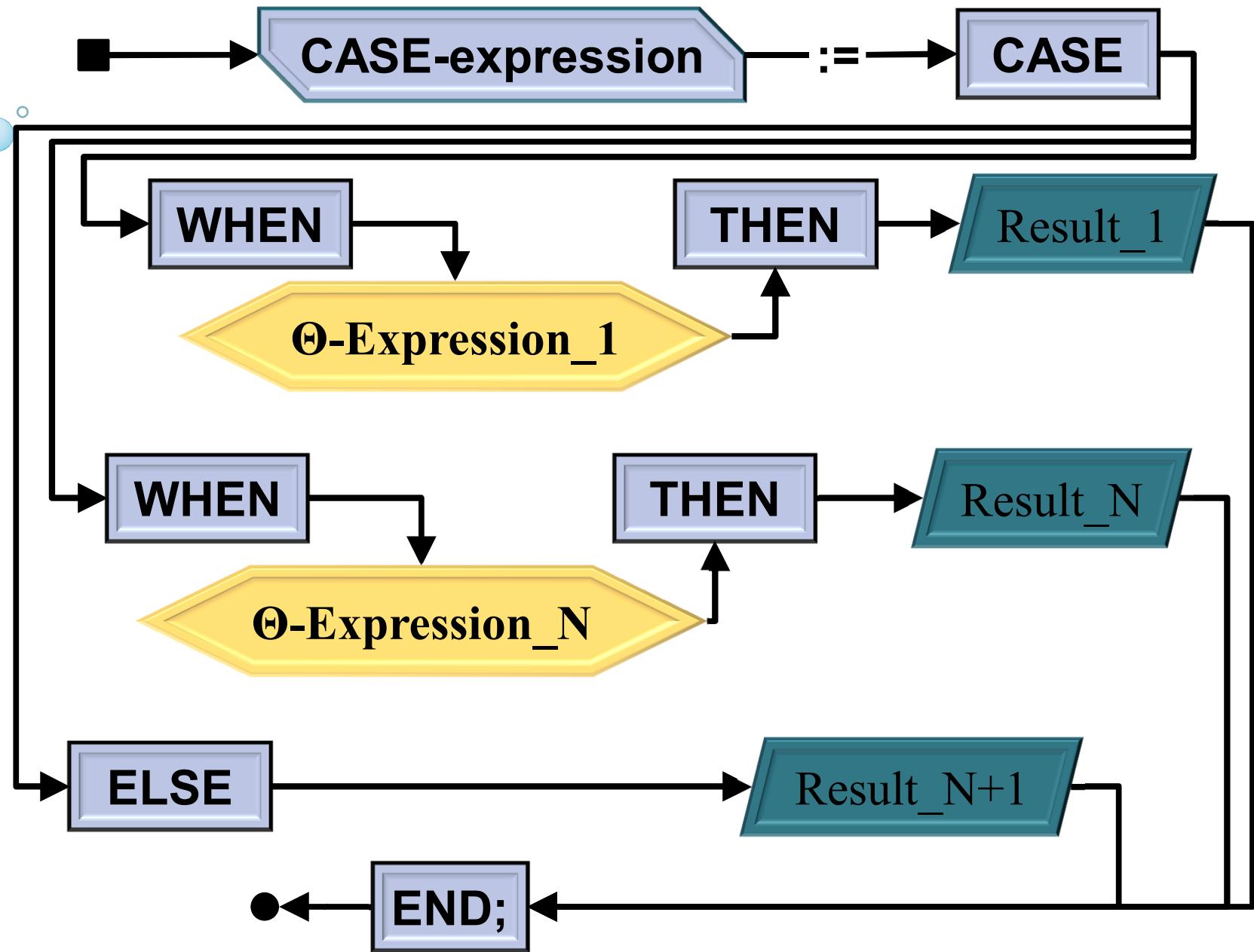
give_bonus(employee_id, 0);

END CASE;

Простий CASE -вираз. Синтаксис.



Пошуковий CASE -вираз. Синтаксис.



CASE -вираз. Приклад.

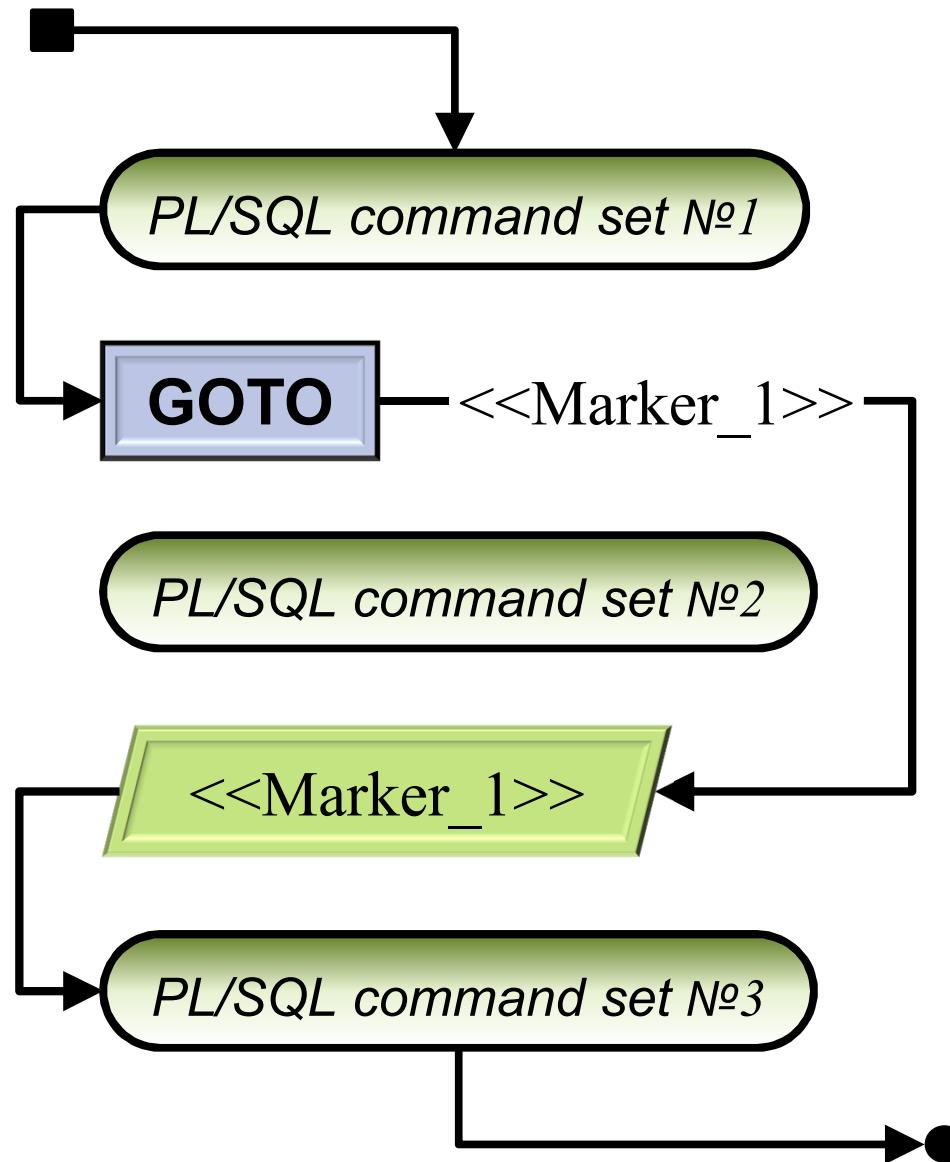
DECLARE

```
    salary NUMBER := 20000;  
    employee_id NUMBER := 36325;  
    bonus_amount NUMBER;
```

BEGIN

```
    bonus_amount := CASE  
        WHEN salary >= 10000 AND salary <= 20000  
        THEN 1500  
        WHEN salary > 20000 AND salary <= 40000  
        THEN 1000  
        WHEN salary > 40000  
        THEN 500  
        ELSE 0  
    END * 10;  
    DBMS_OUTPUT.PUT_LINE(bonus_amount);  
END;
```

Команда GOTO. Синтаксис.



Команда **GOTO**. Приклад.

BEGIN

```
GOTO second_output;  
DBMS_OUTPUT.PUT_LINE  
(‘Цей рядок ніколи не виконується.’);  
<<second_output>>  
DBMS_OUTPUT.PUT_LINE  
(‘Виконується цей рядок!’);
```

END;

- За міткою повинна слідувати хоча б одна виконувана команда *PL/SQL*.
- Цільова мітка повинна бути розташована в межах області дії оператора *GOTO*.
- Цільова мітка повинна бути розташована у тій же частині блоку *PL/SQL*, що і оператор *GOTO*.