Database administration advanced

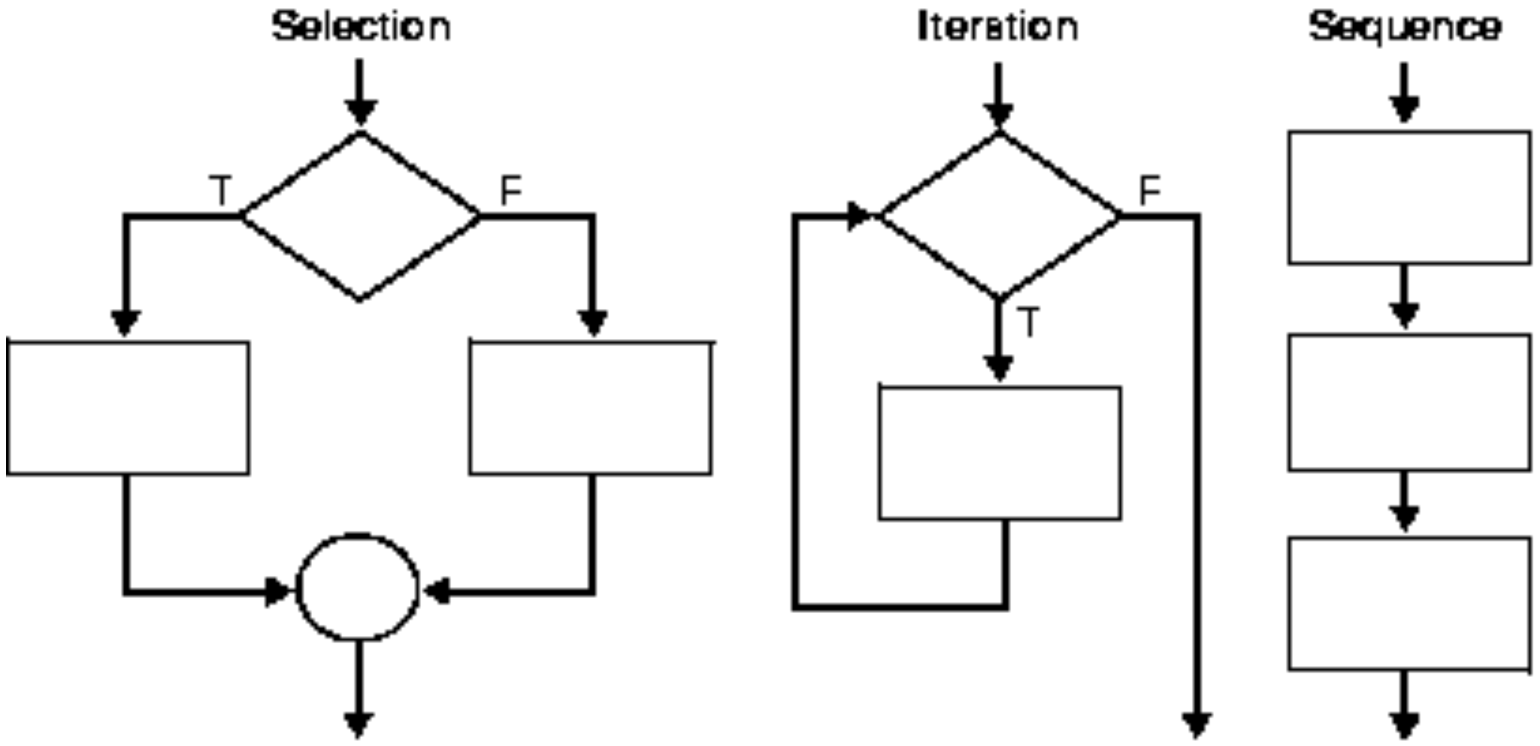
Andres Martínez Gutiérrez. 2018-1

Agenda

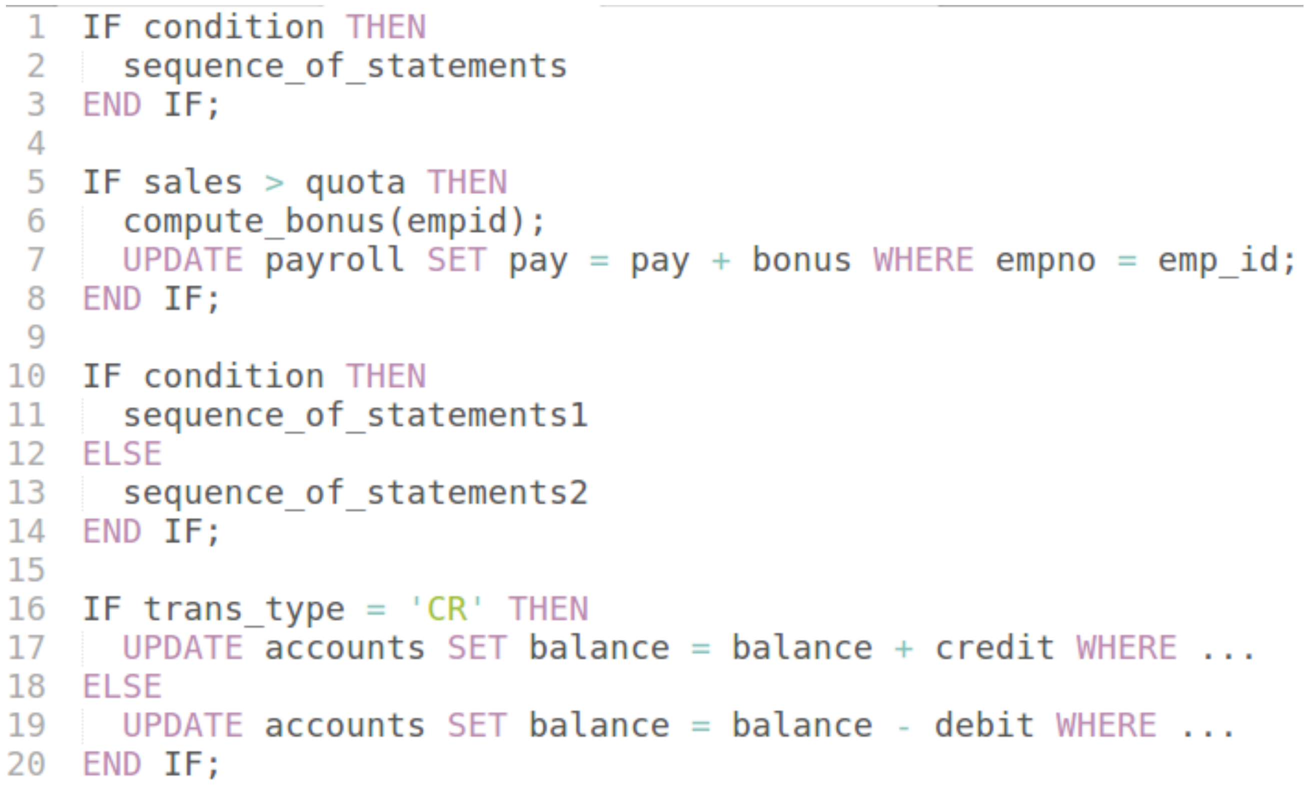
1. PL/SQL Control Structures

PL/SQL Control Structures

https://docs.oracle.com/cd/A97630_01/appdev.920/a96624/04_struc.htm


Selection
T
F
Iteration
F
T
Sequence

Conditional Control: IF and CASE Statements

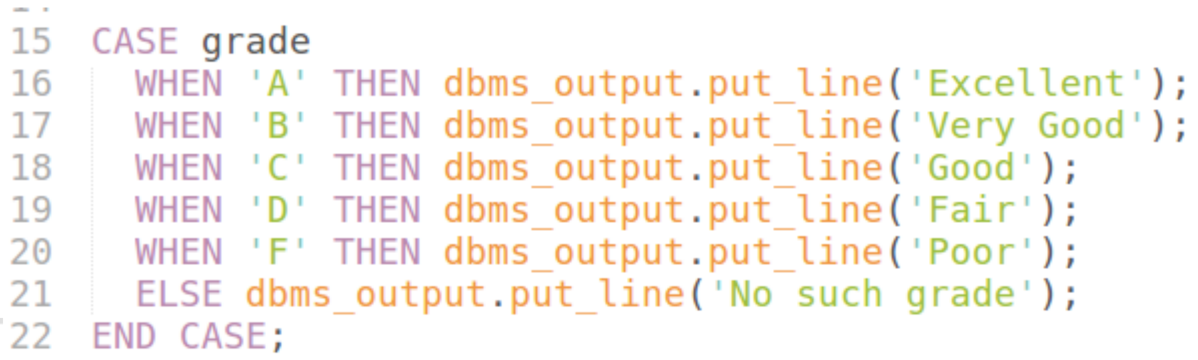


```
IF condition THEN
  sequence_of_statements
END IF;

IF sales > quota THEN
  compute_bonus(empid);
  UPDATE payroll SET pay = pay + bonus WHERE empno = emp_id;
END IF;

IF condition THEN
  sequence_of_statements1
ELSE
  sequence_of_statements2
END IF;

IF trans_type = 'CR' THEN
  UPDATE accounts SET balance = balance + credit WHERE ...
ELSE
  UPDATE accounts SET balance = balance - debit WHERE ...
END IF;
```

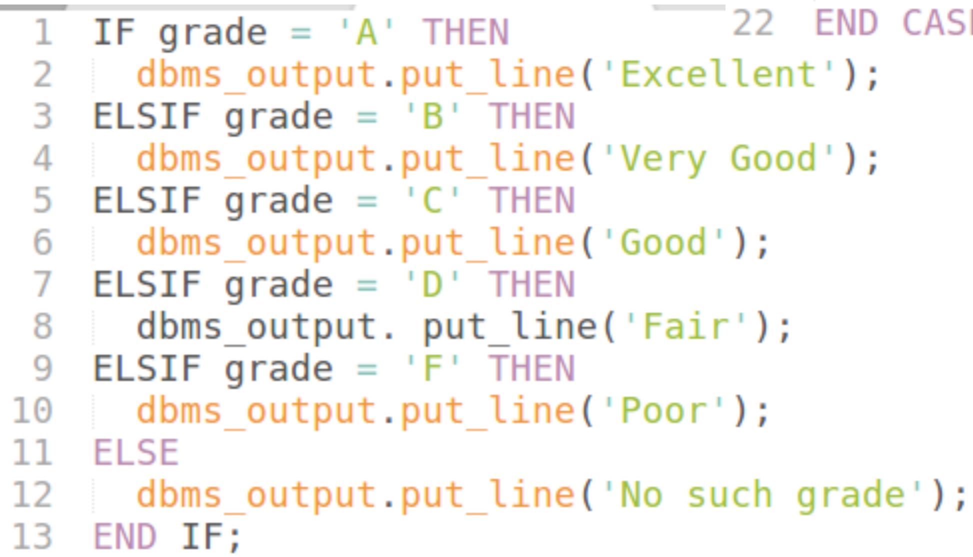
Conditional Control: IF-THEN-ELSIF Statement

```
IF condition1 THEN
  sequence_of_statements1
ELSIF condition2 THEN
  sequence_of_statements2
ELSE
  sequence_of_statements3
END IF;

BEGIN
  ...
  IF sales > 50000 THEN
    bonus := 1500;
  ELSIF sales > 35000 THEN
    bonus := 500;
  ELSE
    bonus := 100;
  END IF;
  INSERT INTO payroll VALUES (emp_id, bonus, ...);
END;
```

Conditional Control: CASE Statement

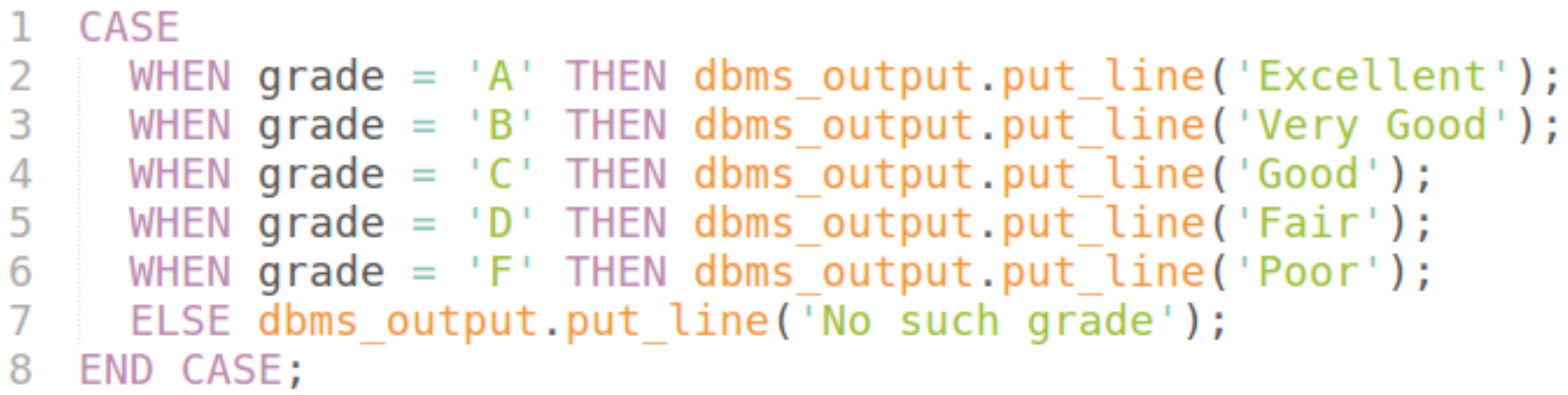


```
CASE grade
  WHEN 'A' THEN dbms_output.put_line('Excellent');
  WHEN 'B' THEN dbms_output.put_line('Very Good');
  WHEN 'C' THEN dbms_output.put_line('Good');
  WHEN 'D' THEN dbms_output.put_line('Fair');
  WHEN 'F' THEN dbms_output.put_line('Poor');
  ELSE dbms_output.put_line('No such grade');
END CASE;
```



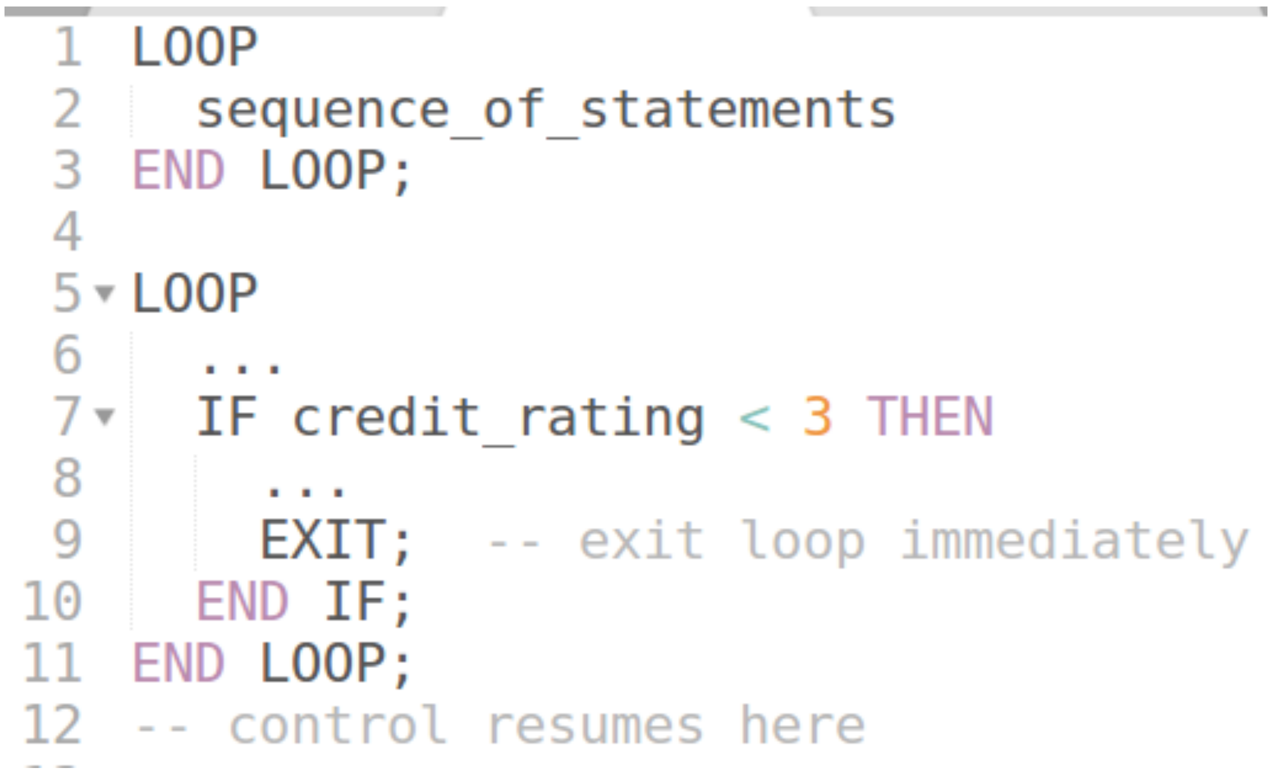
```
IF grade = 'A' THEN
  dbms_output.put_line('Excellent');
ELSIF grade = 'B' THEN
  dbms_output.put_line('Very Good');
ELSIF grade = 'C' THEN
  dbms_output.put_line('Good');
ELSIF grade = 'D' THEN
  dbms_output. put_line('Fair');
ELSIF grade = 'F' THEN
  dbms_output.put_line('Poor');
ELSE
  dbms_output.put_line('No such grade');
END IF;
```

Conditional Control: CASE Statement



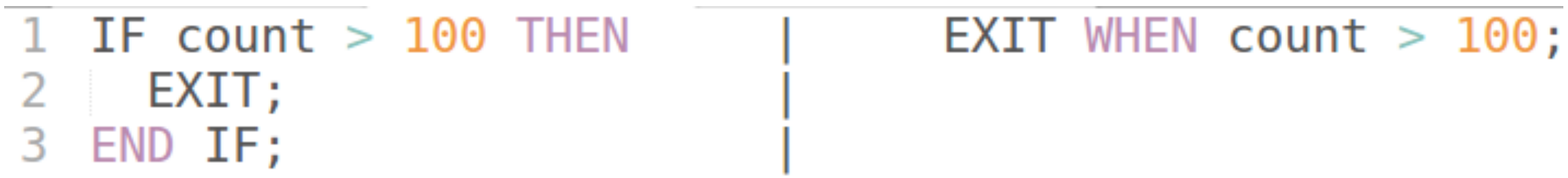
```
CASE
  WHEN grade = 'A' THEN dbms_output.put_line('Excellent');
  WHEN grade = 'B' THEN dbms_output.put_line('Very Good');
  WHEN grade = 'C' THEN dbms_output.put_line('Good');
  WHEN grade = 'D' THEN dbms_output.put_line('Fair');
  WHEN grade = 'F' THEN dbms_output.put_line('Poor');
  ELSE dbms_output.put_line('No such grade');
END CASE;
```

Iterative Control: LOOP and EXIT Statements



```
LOOP
  sequence_of_statements
END LOOP;

LOOP
  ...
  IF credit_rating < 3 THEN
    ...
    EXIT;  -- exit loop immediately
  END IF;
END LOOP;
-- control resumes here
```

Iterative Control: EXIT-WHEN



```
IF count > 100 THEN
  EXIT;
END IF;
```

```
EXIT WHEN count > 100;
```

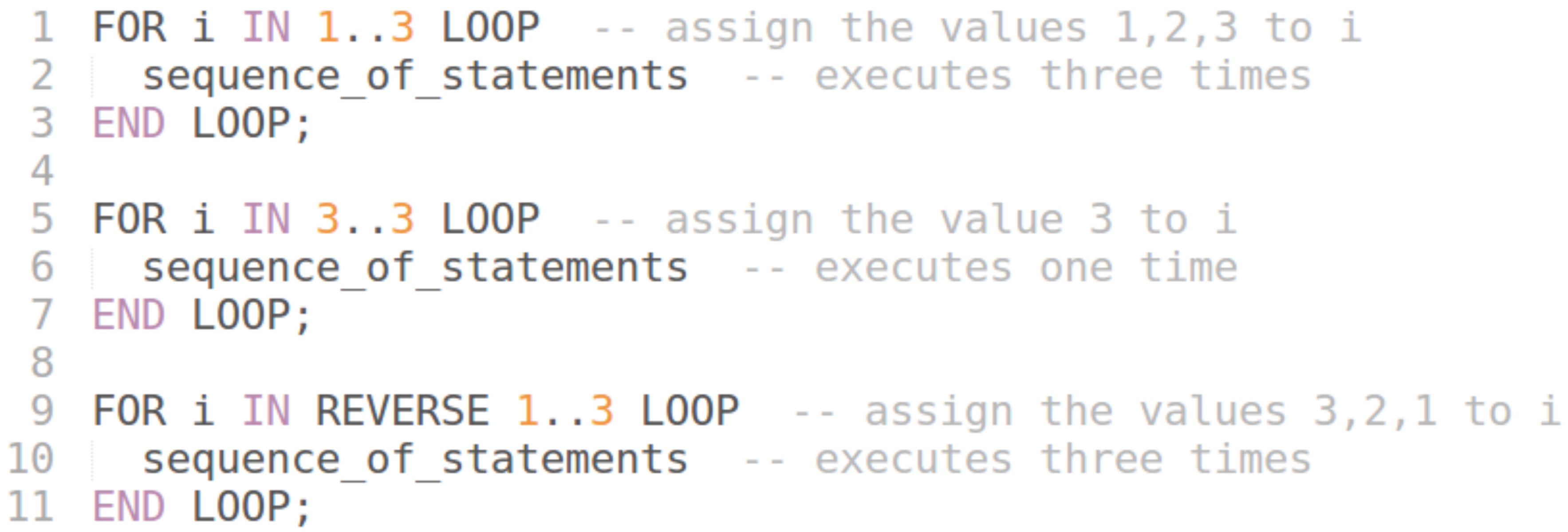
Iterative Control: Loop Labels

```
<<my_loop>>
LOOP
  ...
END LOOP my_loop;

<<outer>>
LOOP
  ...
  LOOP
    ...
    EXIT outer WHEN ...  -- exit both loops
  END LOOP;
  ...
END LOOP outer;
```

Iterative Control: WHILE-LOOP

```
WHILE total <= 25000 LOOP
  ...
  SELECT sal INTO salary FROM emp WHERE ...
  total := total + salary;
END LOOP;

LOOP
  sequence_of_statements
  EXIT WHEN boolean_expression;
END LOOP;

done := FALSE;
WHILE NOT done LOOP
   sequence_of_statements
   done := boolean_expression;
END LOOP;
```

Iterative Control: FOR-LOOP



```
FOR i IN 1..3 LOOP  -- assign the values 1,2,3 to i
  sequence_of_statements  -- executes three times
END LOOP;

FOR i IN 3..3 LOOP  -- assign the value 3 to i
  sequence_of_statements  -- executes one time
END LOOP;

FOR i IN REVERSE 1..3 LOOP  -- assign the values 3,2,1 to i
  sequence_of_statements  -- executes three times
END LOOP;
```

NULL Statement

```
EXCEPTION
  WHEN ZERO_DIVIDE THEN
    ROLLBACK;
  WHEN VALUE_ERROR THEN
    INSERT INTO errors VALUES ...
    COMMIT;
  WHEN OTHERS THEN
    NULL;
END;

IF rating > 90 THEN
  compute_bonus(emp_id);
ELSE
  NULL;
END IF;
```

Thank you.