GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: BCA - IOP

Course Code: BCAS3031

Course Name: PL/SQL & Cursors and Triggers

Dr. T. Poongodi

Associate Professor

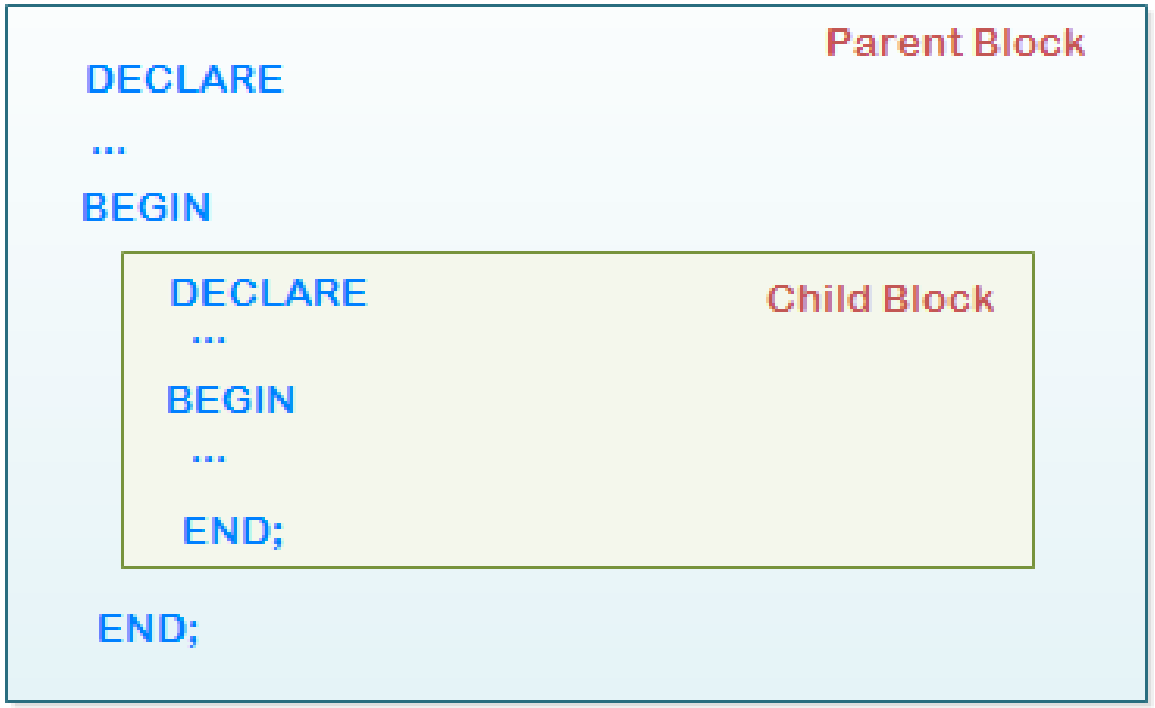
PL/SQL nested block

- PL/SQL block embedded inside another PL/SQL block
- nest a block means to embed one or more PL/SQL blocks inside another PL/SQL block that provide you with a better control over program execution and exception handling

```
SET SERVEROUTPUT ON;
DECLARE
n_emp_id EMPLOYEES.EMPLOYEE_ID%TYPE := &emp_id1;
BEGIN
        DECLARE
                n_emp_id employees.employee_id%TYPE := &emp_id2;
                v_name employees.first_name%TYPE;
        BEGIN
                SELECT first_name
                INTO v_name
                FROM employees
                WHERE employee_id = n_emp_id;

                DBMS_OUTPUT.PUT_LINE('First name of employee ' || n_emp_id || ' is ' || v_name);

                EXCEPTION
                        WHEN no_data_found THEN
                        DBMS_OUTPUT.PUT_LINE('Employee ' || n_emp_id || ' not found');
        END;
END;
/
```



The outer PL/SQL block is called *parent block* or *enclosing block* and the inner PL/SQL block is known as *child block, nested block* or *enclosed block*.

PL/SQL Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulation.

Types of operators –

- Arithmetic operators
- Relational operators
- Comparison operators
- Logical operators
- String operators

Arithmetic Operators

variable A holds 10 and **variable B** holds 5

Operator	Description	Example
+	Adds two operands	A + B will give 15
-	Subtracts second operand from the first	A - B will give 5
*	Multiplies both operands	A * B will give 50
/	Divides numerator by de-numerator	A / B will give 2
**	Exponentiation operator, raises one operand to the power of other	A ** B will give 100000

Relational Operators

- Relational operators compare two expressions or values and return a Boolean result.
- **variable A** holds 10 and **variable B** holds 20

Operator	Description	Example
=	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A = B) is not true.
!= <> ~=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.

Operator	Description	Example
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true

Comparison Operators

- Comparison operators are used for comparing one expression to another. The result is always either **TRUE, FALSE** or **NULL**.

Operator	Description	Example
LIKE	The LIKE operator compares a character, string, or CLOB value to a pattern and returns TRUE if the value matches the pattern and FALSE if it does not.	If 'Zara Ali' like 'Z% A_i' returns a Boolean true, whereas, 'Nuha Ali' like 'Z% A_i' returns a Boolean false.
BETWEEN	The BETWEEN operator tests whether a value lies in a specified range. x BETWEEN a AND b means that x >= a and x <= b.	If x = 10 then, x between 5 and 20 returns true, x between 5 and 10 returns true, but x between 11 and 20 returns false.
IN	The IN operator tests set membership. x IN (set) means that x is equal to any member of set.	If x = 'm' then, x in ('a', 'b', 'c') returns Boolean false but x in ('m', 'n', 'o') returns Boolean true.
IS NULL	The IS NULL operator returns the BOOLEAN value TRUE if its operand is NULL or FALSE if it is not NULL. Comparisons involving NULL values always yield NULL.	If x = 'm', then 'x is null' returns Boolean false.

Logical Operators

- Logical operators work on Boolean operands and produce Boolean results.
- **variable A** holds true and **variable B** holds false

Operator	Description	Examples
and	Called the logical AND operator. If both the operands are true then condition becomes true.	(A and B) is false.
or	Called the logical OR Operator. If any of the two operands is true then condition becomes true.	(A or B) is true.
not	Called the logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true then Logical NOT operator will make it false.	not (A and B) is true.

PL/SQL Operator Precedence

- Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator.
- For example, **x = 7 + 3 * 2**; here, **x** is assigned **13**, not 20 because operator * has higher precedence than +, so it first gets multiplied with **3*2** and then adds into **7**.
- Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.
- The precedence of operators goes as follows: =, <, >, <=, >=, <>, !=, ~=, ^=, IS NULL, LIKE, BETWEEN, IN.

Operator	Operation
**	exponentiation
+, -	identity, negation
*, /	multiplication, division
+, -, ||	addition, subtraction, concatenation
comparison	
NOT	logical negation
AND	conjunction
OR	inclusion

```sql
SELECT * from student WHERE sname LIKE 'J%';
```

```sql
SELECT * from student WHERE sname LIKE '_ _ _ a';
```

```sql
SELECT * from student WHERE age BETWEEN 12 AND 18;
```

```sql
SELECT * from student WHERE city IN ('Delhi','Goa','Kerela');
```

```sql
SELECT * from student WHERE age IS NULL;
```


Thank You