20-06-2024 Writing Executable Statements

Commenting Code

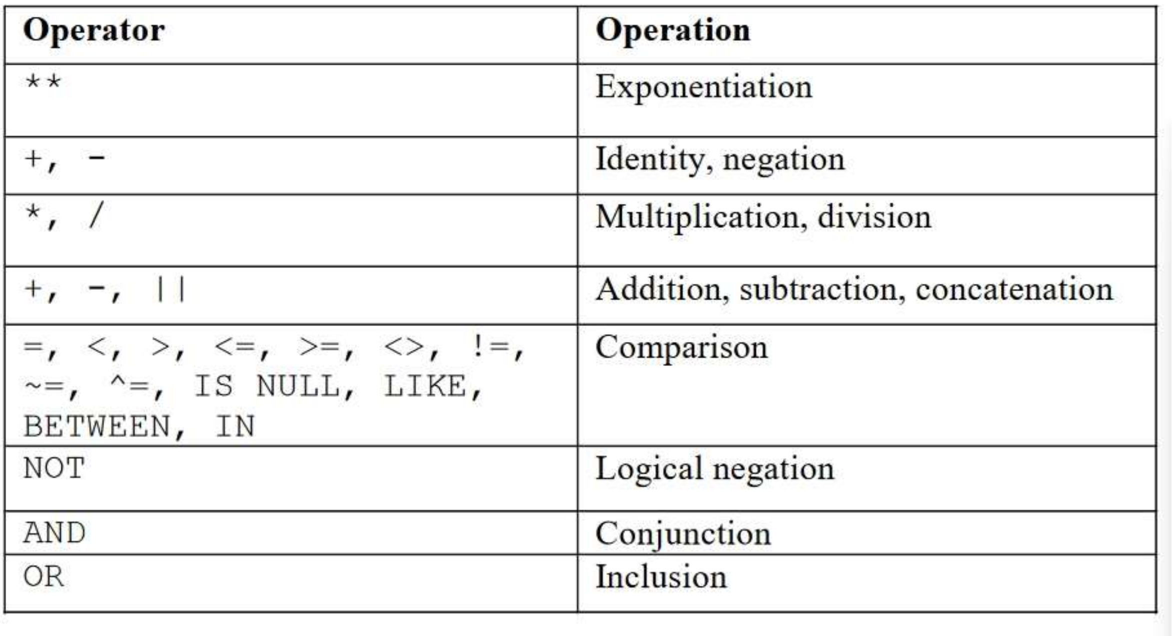
```
1.DECLARE
v_sal NUMBER (9,2);
BEGIN
/* Compute the annual salary based on the
monthly salary input from the user */
v_sal := :g_monthly_sal * 12;
END; -- This is the end of the block
```

SQL Functions in PL/SQL

```
2. v_mailing_address := v_name||CHR(10)|| v_address||CHR(10)||v_state||
CHR(10)||v_zip;
3. v_ename := LOWER(v_ename);
```

Data type Conversion

```
4.v_date := TO_DATE (’January 13, 2001’, ’Month DD, YYYY’);
```



Operator	Operation
`**`	Exponentiation
`+, -`	Identity, negation
`*, /`	Multiplication, division
`+, -, ||`	Addition, subtraction, concatenation
`=, <, >, <=, >=, <>, !=, ~=, ^=, IS NULL, LIKE, BETWEEN, IN`	Comparison
`NOT`	Logical negation
`AND`	Conjunction
`OR`	Inclusion

SELECT Statements in PL/SQL

```
5.DECLARE
v_deptno NUMBER(4);
```

```
v_location_id NUMBER(4);
BEGIN
SELECT department_id, location_id
INTO v_deptno, v_location_id
FROM departments
WHERE department_name = ’Sales’;
END;
```

Retrieving Data in PL/SQL

```
6.SET SERVEROUTPUT ON
DECLARE
v_sum_sal NUMBER(10,2);
v_deptno NUMBER NOT NULL := 60;
BEGIN
SELECT SUM(salary) -- group function
INTO v_sum_sal
FROM employees
WHERE department_id = v_deptno;
DBMS_OUTPUT.PUT_LINE (’The sum salary is ’ ||
TO_CHAR(v_sum_sal));
END;
```

Naming Convention

```
7.DECLARE
hire_date employees.hire_date%TYPE;
sysdate hire_date%TYPE;
employee_id employees.employee_id%TYPE := 176;
BEGIN
SELECT hire_date, sysdate
INTO hire_date, sysdate
FROM employees
WHERE employee_id = employee_id; END;
```