

EXERCISE 5: PL/SQL BASICS

NAME: JAGTAP MAHESH

REG NO. 24MCS1017

Answer all the Questions:

1. Write a simple PL/SQL Block to print Hello World.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello, World!');
END;
/
```

```
PL/SQL procedure successfully completed.

SQL> set serveroutput on;
SQL> BEGIN
2     DBMS_OUTPUT.PUT_LINE('Hello, World!');
3 END;
4 /
Hello, World!

PL/SQL procedure successfully completed.

SQL> |
```

2. Write a simple PL/SQL block to accept two numbers from the user and perform all four basic mathematical operations.

```
DECLARE
    num1 NUMBER;
    num2 NUMBER;
    result NUMBER;
BEGIN
    num1 := &num1;
    num2 := &num2;

    result := num1 + num2;
```

```
dbms_output.put_line('Sum: ' || result);

result := num1 - num2;
dbms_output.put_line('Difference: ' || result);

result := num1 * num2;
dbms_output.put_line('Product: ' || result);

IF num2 != 0 THEN
    result := num1 / num2;
    dbms_output.put_line('Quotient: ' || result);
ELSE
    dbms_output.put_line('Cannot divide by zero.');
```

END IF;

END;

/

```

SQL> DECLARE
2     num1 NUMBER;
3     num2 NUMBER;
4     result NUMBER;
5 BEGIN
6     num1 := &num1;
7     num2 := &num2;
8
9     result := num1 + num2;
10    dbms_output.put_line('Sum: ' || result);
11
12    result := num1 - num2;
13    dbms_output.put_line('Difference: ' || result);
14
15    result := num1 * num2;
16    dbms_output.put_line('Product: ' || result);
17
18    IF num2 != 0 THEN
19        result := num1 / num2;
20        dbms_output.put_line('Quotient: ' || result);
21    ELSE
22        dbms_output.put_line('Cannot divide by zero.');
```

23 END IF;
24 END;
25 /
Enter value for num1: 8
old 6: num1 := &num1;
new 6: num1 := 8;
Enter value for num2: 5
old 7: num2 := &num2;
new 7: num2 := 5;
Sum: 13
Difference: 3
Product: 40
Quotient: 1.6

3. Write a simple PL/SQL block to fetch the salary of an employee named 'Ravi' from the Employee table.

```

CREATE TABLE Employee (
    emp_id NUMBER PRIMARY KEY,
    name VARCHAR2(50),
    salary NUMBER
);

```

```

INSERT INTO Employee (emp_id, name, salary) VALUES (1, 'Ravi', 50000);
INSERT INTO Employee (emp_id, name, salary) VALUES (2, 'Ajay', 60000);
INSERT INTO Employee (emp_id, name, salary) VALUES (3, 'Akshay', 55000);

```

```

SET SERVEROUTPUT ON;

```

```

DECLARE
    v_salary NUMBER;
BEGIN
    SELECT salary
    INTO   v_salary
    FROM   Employee
    WHERE  name = 'Ravi';

    DBMS_OUTPUT.PUT_LINE('The salary of Ravi is: ' || TO_CHAR(v_salary));
END;
/

```

```

SQL> DECLARE
2     v_salary  NUMBER;
3 BEGIN
4     SELECT salary
5     INTO   v_salary
6     FROM   Employee
7     WHERE  name = 'Ravi';
8
9     DBMS_OUTPUT.PUT_LINE('The salary of Ravi is: ' || TO_CHAR(v_salary));
10 END;
11 /
The salary of Ravi is: 50000

PL/SQL procedure successfully completed.

```

4. Write a simple PL/SQL block to accept the department number from the user and print the count of employees in that department specified by the user.

```

CREATE TABLE Employee (
    emp_id  NUMBER PRIMARY KEY,
    name    VARCHAR2(50),
    salary  NUMBER,
    dept_id NUMBER
);

```

```

INSERT INTO Employee (emp_id, name, salary, dept_id) VALUES (1, 'Ravi',
50000, 10);
INSERT INTO Employee (emp_id, name, salary, dept_id) VALUES (2, 'ajay',
60000, 20);
INSERT INTO Employee (emp_id, name, salary, dept_id) VALUES (3,
'Akshay', 55000, 10);

```

```
INSERT INTO Employee (emp_id, name, salary, dept_id) VALUES (4, 'Aditya',
52000, 30);
```

```
INSERT INTO Employee (emp_id, name, salary, dept_id) VALUES (5, 'Rahul',
48000, 20);
```

```
SQL> select * from employee;
```

EMP_ID	NAME	SALARY	DEPT_ID
1	Ravi	50000	10
2	ajay	60000	20
3	Akshay	55000	10
4	Aditya	52000	30
5	Rahul	48000	20

```
DECLARE
```

```
    v_deptno NUMBER;
```

```
    v_count  NUMBER;
```

```
BEGIN
```

```
    v_deptno := &deptno;
```

```
    SELECT COUNT(*)
```

```
    INTO v_count
```

```
    FROM Employee
```

```
    WHERE dept_id = v_deptno;
```

```
    DBMS_OUTPUT.PUT_LINE('The number of employees in department ' ||
TO_CHAR(v_deptno) || ' is: ' || TO_CHAR(v_count));
```

```
END;
```

```
/
```

```
SQL> DECLARE
2     v_deptno  NUMBER;
3     v_count   NUMBER;
4 BEGIN
5     v_deptno := &deptno;
6     SELECT COUNT(*)
7     INTO v_count
8     FROM Employee
9     WHERE dept_id = v_deptno;
10
11     DBMS_OUTPUT.PUT_LINE('The number of employees in department ' || TO_CHAR(v_deptno) || ' is: ' || TO_CH
AR(v_count));
12 END;
13 /
Enter value for deptno: 10
old 5: v_deptno := &deptno;
new 5: v_deptno := 10;
The number of employees in department 10 is: 2
```

5. Write a simple PL/SQL block to get the age of a person as input and determine if the person is eligible to vote or not.

```
DECLARE
```

```

    age NUMBER;
BEGIN

    age := &age;

    IF age >= 18 THEN
        dbms_output.put_line('The person is eligible to vote.');
```

```
    ELSE
```

```
        dbms_output.put_line('The person is not eligible to vote.');
```

```
    END IF;
```

```
END;
```

```
/
```

```

Enter value for age: 20
old   5:      age := &age;
new   5:      age := 20;
The person is eligible to vote.
```

```

Enter value for age: 17
old   5:      age := &age;
new   5:      age := 17;
The person is not eligible to vote.
```

6. Write a simple PL/SQL block to find the greatest of three given numbers.

```
DECLARE
```

```
    a NUMBER;
```

```
    b NUMBER;
```

```
    c NUMBER;
```

```
BEGIN
```

```
    a := &a_input;
```

```
    b := &b_input;
```

```
    c := &c_input;
```

```
    IF a > b AND a > c THEN
```

```
        dbms_output.Put_line('Greatest number is ' || a);
```

```
    ELSIF b > a AND b > c THEN
```

```
        dbms_output.Put_line('Greatest number is ' || b);
```

```
    ELSE
```

```

        dbms_output.Put_line('Greatest number is ' || c);
    END IF;
END;
/

```

```

Enter value for a_input: 23
old 6:      a := &a_input;
new 6:      a := 23;
Enter value for b_input: 12
old 7:      b := &b_input;
new 7:      b := 12;
Enter value for c_input: 4
old 8:      c := &c_input;
new 8:      c := 4;
Greatest number is 23

```

7. Write a simple PL/SQL block to get the day of the week as input from the user and print the corresponding day. 1 –SUN and 7-SAT

```

DECLARE
    user_day NUMBER;
    day_name VARCHAR2(10);
BEGIN
    user_day := &user_input;
    CASE user_day
        WHEN 1 THEN day_name := 'SUN';
        WHEN 2 THEN day_name := 'MON';
        WHEN 3 THEN day_name := 'TUE';
        WHEN 4 THEN day_name := 'WED';
        WHEN 5 THEN day_name := 'THU';
        WHEN 6 THEN day_name := 'FRI';
        WHEN 7 THEN day_name := 'SAT';
        ELSE day_name := 'Invalid day';
    END CASE;
    DBMS_OUTPUT.PUT_LINE('Day of the week: ' || day_name);
END;
/

```

```
Enter value for user_input: 1
old 5:      user_day := &user_input;
new 5:      user_day := 1;
Day of the week: SUN

PL/SQL procedure successfully completed.
```

```
Enter value for user_input: 2
old 5:      user_day := &user_input;
new 5:      user_day := 2;
Day of the week: MON

PL/SQL procedure successfully completed.
```

9. Write a simple PL/SQL block to print the sum of first n natural numbers using a for loop.

```
DECLARE
    n NUMBER;
    sum NUMBER;
BEGIN
    n := &n;
    sum := 0;
    FOR i IN 1 .. n LOOP
        sum := sum + i;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('The sum of the first ' || TO_CHAR(n) || '
natural numbers is: ' || TO_CHAR(sum));
END;
/
```


STDIN

5

Output:

```
Enter value for n: old 5:      n      := &n;
new 5:      n      := 5;
The sum of the first 5 natural numbers is: 15
```

↓ 72.0 Kbps
↑ 0.4 Kbps

10. Write a simple PL/SQL block to print the sum of first n natural numbers using a while loop.

```
DECLARE
    n    NUMBER := &n;
    sum  NUMBER := 0;
    i    NUMBER := 1;
BEGIN
    WHILE i <= n LOOP
        sum := sum + i;
        i := i + 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('The sum of the first ' || TO_CHAR(n) || '
natural numbers is: ' || TO_CHAR(sum));
END;
/
```

STDIN

20

Output:

```
Enter value for n: old 2:      n          NUMBER := &n;  
new 2:      n          NUMBER := 20;  
The sum of the first 20 natural numbers is: 210
```

↓ 0.0 Kbps
↑ 0.0 Kbps