PL/SQL LAB ASSIGNMENTS

NAME : Vashuki Achari
ROLL : 102103503
CLASS : 2CO18

LAB ASSIGNMENT 1

ANSWERS 1 :

```
DECLARE
num1 NUMBER := 10; num2 NUMBER := 20; num3 NUMBER := 15; greatest
NUMBER;
BEGIN
IF num1 > num2 AND num1 > num3 THEN
greatest := num1;
ELSIF num2 > num3 THEN
greatest := num2; ELSE
greatest := num3; END IF;
dbms_output.put_line('The greatest number is ' || greatest); END;
```

```
Statement processed.
The greatest number is 20
```

ANSWERS 2 :

```
DECLARE
num NUMBER := 45; BEGIN
IF MOD(num, 2) = 0 THEN dbms_output.put_line(num || ' is even');
ELSE
dbms_output.put_line(num || ' is odd');
END IF; END;
```

```
Statement processed.
45 is odd
```

ANSWERS 3 :

```
DECLARE
```

```
marks NUMBER := 45; BEGIN
IF marks>80 THEN dbms_output.put_line(marks|| ' is A');
ELSIF marks >70 AND marks<80 THEN dbms_output.put_line(marks || ' is B');
ELSIF marks >60 AND marks<70 THEN dbms_output.put_line(marks || ' is C');
ELSIF marks >50 AND marks<60 THEN dbms_output.put_line(marks || ' is D');
ELSIF marks >40 AND marks<50 THEN dbms_output.put_line(marks || ' is E');
END IF; END;
```

```
Statement processed.
45 is E
```

ANSWERS 4 :

```
DECLARE
num NUMBER := 55; BEGIN
FOR i IN 1..10 LOOP
dbms_output.put_line(num || ' x ' || i || ' = ' || num*i);
END LOOP; END;
```

```
Statement processed.
55 x 1 = 55
55 x 2 = 110
55 x 3 = 165
55 x 4 = 220
55 x 5 = 275
55 x 6 = 330
55 x 7 = 385
55 x 8 = 440
55 x 9 = 495
55 x 10 = 550
```

ANSWERS 5 :

```
DECLARE
num NUMBER := 5; factorial NUMBER := 1; i NUMBER := 1;
BEGIN
WHILE i <= num LOOP
factorial := factorial * i;
i := i + 1;
END LOOP;
dbms_output.put_line('Factorial of ' || num || ' is ' || factorial);
```

END;

```
Statement processed.
Factorial of 5 is 120
```

ANSWERS 6 :

```
DECLARE
n NUMBER := 55;
fib_prev NUMBER := 0;
fib_curr NUMBER := 1;
fib_next NUMBER;
BEGIN
IF n = 0 THEN
dbms_output.put_line('The 0-th Fibonacci number is 0');
ELSIF n = 1 THEN
dbms_output.put_line('The 1-st Fibonacci number is 1');
ELSE
FOR i IN 2..n LOOP
fib_next := fib_prev + fib_curr; fib_prev := fib_curr;
fib_curr := fib_next;
END LOOP;
dbms_output.put_line('The ' || n || '-th Fibonacci number is ' || fib_curr);
END IF;
END;
```

```
Statement processed.
The 55-th Fibonacci number is 139583862445
```

ANSWERS 7 :

```
DECLARE
num NUMBER := 12345; rev NUMBER := 0;
rem NUMBER;
BEGIN
WHILE num > 0 LOOP
rem := num MOD 10; rev := rev * 10 + rem; num := floor(num / 10);
END LOOP;
DBMS_OUTPUT.PUT_LINE('The reverse of the number is: ' || TO_CHAR(rev));
EXCEPTION
WHEN OTHERS THEN dbms_output.put_line('Error: ' || SQLERRM);
END; /
```

```
Statement processed.
The reverse of the number is: 54321
```

ANSWERS 8 :

```
DECLARE
num1 NUMBER; num2 NUMBER; choice NUMBER; result NUMBER;
BEGIN
dbms_output.put_line('Enter first number: ');
num1 := 55;
dbms_output.put_line('Enter second number: ');
num2 := 12;
dbms_output.put_line('Enter choice: 1. Addition 2. Subtraction 3. Multiplication
4.
Division');
choice := 3;
IF choice = 1 THEN
result := num1 + num2; ELSIF choice = 2 THEN
result := num1 - num2; ELSIF choice = 3 THEN
result := num1 * num2; ELSIF choice = 4 THEN
result := num1 / num2; ELSE
dbms_output.put_line('Invalid choice!'); END IF;
dbms_output.put_line('Result is: ' || result); END;
```

```
Statement processed.
Enter first number:
Enter second number:
Enter choice: 1. Addition 2. Subtraction 3. Multiplication 4. Division
Result is: 660
```

ANSWERS 9 :

```
DECLARE
num NUMBER := 5; BEGIN
FOR i IN 1..4 LOOP dbms_output.put_line(num); num := num + 5;
END LOOP; END;
/
```

```
Statement processed.
5
10
15
20
```

ANSWERS 10 :

```
DECLARE
cur_time VARCHAR2(20); BEGIN
cur_time := TO_CHAR(SYSDATE, 'HH24:MI:SS');
IF cur_time >= '00:00:00' AND cur_time < '12:00:00' THEN
DBMS_OUTPUT.PUT_LINE('Good Morning!');
ELSIF cur_time >= '12:00:00' AND cur_time < '18:00:00' THEN
DBMS_OUTPUT.PUT_LINE('Good Afternoon!');
ELSE
DBMS_OUTPUT.PUT_LINE('Good Night!');
END IF; END;
/
```

```
Statement processed.
Good Afternoon!
```

LAB ASSIGNMENT 2

ANSWERS 1 :

```
CREATE TABLE EMP(
ename varchar(20), emp_id int
)
INSERT into EMP values('tejas',55); INSERT into EMP values('vashhuu',22);
DECLARE
temp varchar(20);
BEGIN
```

SELECT ename into temp from EMP where ename='puru';
exception
WHEN no_data_found THEN
dbms_output.put_line('ERROR'); dbms_output.put_line('there is no name as');
dbms_output.put_line('puru in EMP table');
end;

```
Statement processed.
ERROR
there is no name as
puru in EMP table
```

Too many rows DECLARE
temp varchar(20); BEGIN
-- raises an exception as SELECT
-- into trying to return too many rows
SELECT ename into temp from EMP; dbms_output.put_line(temp);
EXCEPTION
WHEN too_many_rows THEN
dbms_output.put_line('error trying to SELECT too many rows'); end;

```
Statement processed.
error trying to SELECT too many rows
```

ANSWERS 2 :
Write a PL/SQL code to display a message to check whether the record is deleted or not.
ANS-DECLARE
v_count NUMBER; BEGIN
-- Delete a record from the table
DELETE FROM EMP WHERE emp_id = 22;
-- Check if the record was deleted
SELECT COUNT(*) INTO v_count FROM EMP WHERE emp_id = 22;
-- Display a message based on the result IF v_count = 0 THEN
DBMS_OUTPUT.PUT_LINE('Record deleted successfully.');
ELSE
DBMS_OUTPUT.PUT_LINE('Record could not be deleted.');
END IF;
END;

```
Statement processed.
Record deleted successfully.
```

ANSWERS 3 :

```
CREATE TABLE EMP(
ename varchar(20), empno int,
job varchar(20), salary int,
deptno int )
INSERT into EMP values('tejas',55,'mistri',100000,1); I
NSERT into EMP values('vashhuu',22,'jingjang',70000,2);
INSERT into EMP values('puru',33,'civil',10000,3);
DECLARE
v_count NUMBER; BEGIN
-- Delete records from the table
DELETE FROM EMP WHERE salary > 50000;
-- Get the number of records deleted v_count := SQL%ROWCOUNT;
-- Display a message with the count of records deleted IF v_count = 0 THEN
DBMS_OUTPUT.PUT_LINE('No records were deleted.');
ELSE
DBMS_OUTPUT.PUT_LINE(v_count || ' records were deleted.');
END IF;
END;
```

```
Statement processed.
2 records were deleted.
```

ANSWERS 4 :

```
CREATE TABLE EMP( ename varchar(20), empno int,
job varchar(20), salary int, deptno int )
INSERT into EMP values('tejas',55,'mistri',100000,1);
INSERT into EMP values('vashhuu',22,'jingjang',70000,2);
INSERT into EMP values('puru',33,'civil',10000,3);
-- DECLARE
-- Declare variables using %TYPE empno_var EMP.empno%TYPE;
ename_var EMP.ename%TYPE; job_var EMP.job%TYPE;
sal_var EMP.salary%TYPE; deptno_var EMP.deptno%TYPE;
-- Declare record variable using %ROWTYPE emp_rec EMP%ROWTYPE;
BEGIN
-- Fetch all employee records from the EMP table FOR emp_rec IN (SELECT *
FROM EMP) LOOP
-- Assign values to variables using %ROWTYPE empno_var := emp_rec.empno;
```

```
ename_var := emp_rec.ename;
job_var := emp_rec.job;
sal_var := emp_rec.salary; deptno_var := emp_rec.deptno;
-- Display employee details DBMS_OUTPUT.PUT_LINE('Employee No: ' ||
empno_var);
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || ename_var);
DBMS_OUTPUT.PUT_LINE('Job: ' || job_var);
DBMS_OUTPUT.PUT_LINE('Salary: ' || sal_var);
DBMS_OUTPUT.PUT_LINE('Department No: ' || deptno_var);
DBMS_OUTPUT.PUT_LINE('----------------------');
END LOOP;
END; /
```

```
Statement processed.
Employee No: 55
Employee Name: tejas
Job: mistri
Salary: 100000
Department No: 1
----------------------
Employee No: 22
Employee Name: vashhuu
Job: jingjang
Salary: 70000
Department No: 2
----------------------
Employee No: 33
Employee Name: puru
Job: civil
Salary: 10000
Department No: 3
----------------------
```

ANSWERS 5 :

```
CREATE TABLE EMP(
ename varchar(20), empno int,
job varchar(20), salary int,
deptno int )
INSERT into EMP values('tejas',55,'mistri',100000,10);
INSERT into EMP values('vashhuu',22,'jingjang',70000,10); I
NSERT into EMP values('puru',33,'civil',10000,3);
```

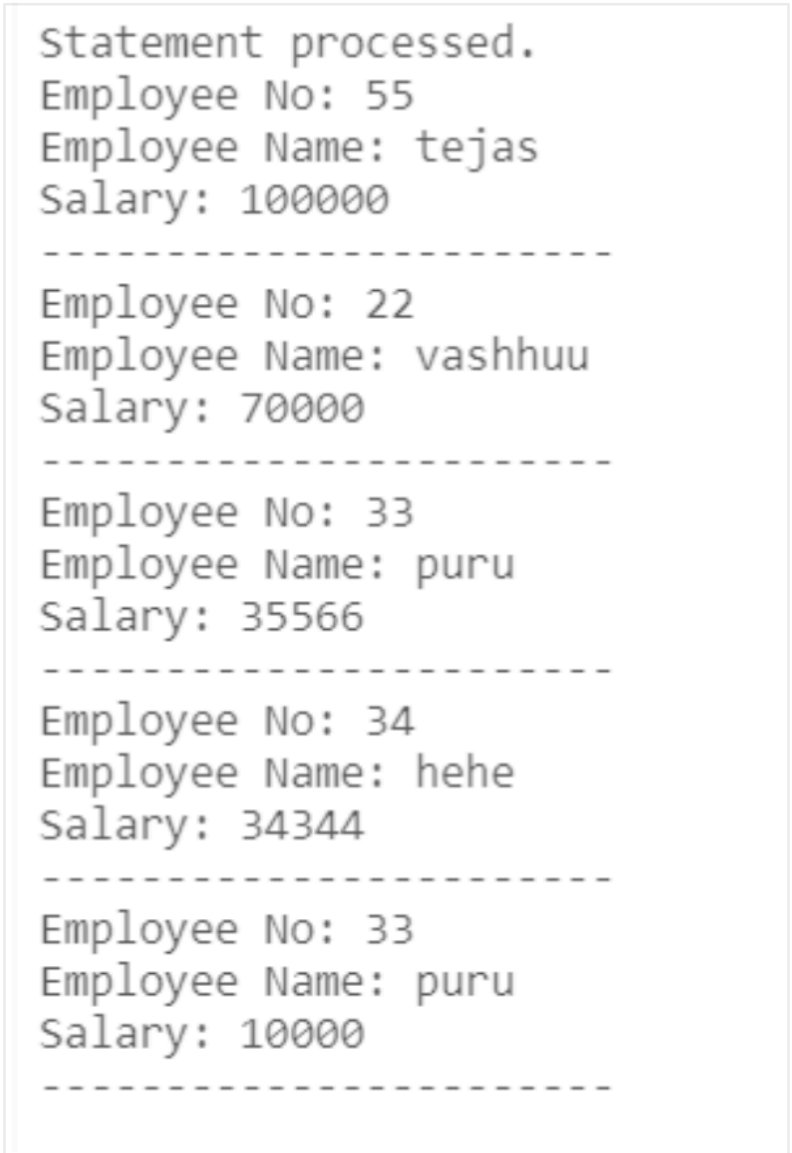
```
DECLARE
-- Declare variables for employee details empno_var EMP.empno%TYPE;
ename_var EMP.ename%TYPE; job_var EMP.job%TYPE;
-- Declare cursor for employees in department 10 CURSOR dept_10_cur IS
SELECT EMPNO, ENAME, JOB FROM EMP
WHERE deptno = 10;
BEGIN
-- Open the cursor and loop through the employee records OPEN dept_10_cur;
LOOP
FETCH dept_10_cur INTO empno_var, ename_var, job_var;
EXIT WHEN dept_10_cur%NOTFOUND;
-- Display employee details DBMS_OUTPUT.PUT_LINE('Employee No: ' ||
empno_var);
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || ename_var);
DBMS_OUTPUT.PUT_LINE('Job: ' || job_var);
DBMS_OUTPUT.PUT_LINE('----------------------');
END LOOP;
CLOSE dept_10_cur; END;
/
```

```
Statement processed.
Employee No: 55
Employee Name: tejas
Job: mistri
------------------------
Employee No: 22
Employee Name: vashhuu
Job: jingjang
------------------------
```

ANSWERS 6 :

```
CREATE TABLE EMP(
ename varchar(20), empno int,
job varchar(20), salary int,
deptno int )
INSERT into EMP values('tejas',55,'mistri',100000,10);
INSERT into EMP values('vashhuu',22,'jingjang',70000,10);
INSERT into EMP values('puru',33,'civil',10000,3);
INSERT into EMP values('hehe',34,'jhumri',34344,5);
INSERT into EMP values('puru',33,'civil',35566,5);
INSERT into EMP values('puru',33,'civil',2424,2);
DECLARE
-- Declare variables for employee details empno_var EMP.empno%TYPE;
```

```
ename_var EMP.ename%TYPE; sal_var EMP.salary%TYPE;
-- Declare cursor for top 5 highest-paid employees CURSOR top_5_cur IS
SELECT empno, ename, salary FROM EMP
ORDER BY salary DESC FETCH FIRST 5 ROWS ONLY;
BEGIN
-- Open the cursor and loop through the employee records OPEN top_5_cur;
LOOP
FETCH top_5_cur INTO empno_var, ename_var, sal_var;
EXIT WHEN top_5_cur%NOTFOUND;
-- Display employee details DBMS_OUTPUT.PUT_LINE('Employee No: ' ||
empno_var);
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || ename_var);
DBMS_OUTPUT.PUT_LINE('Salary: ' || sal_var);
DBMS_OUTPUT.PUT_LINE('----------------------');
END LOOP;
CLOSE top_5_cur; END;
/
```



ANSWERS 7 :

```
DECLARE
-- Declare variables for total salary and record count
total_sal NUMBER := 0;
rec_count NUMBER := 0;
sal_var EMP.salary%TYPE;
-- Declare the sal_var variable with the same
data type as the SAL column
-- Declare cursor for first n records of EMP table CURSOR emp_cur (n
NUMBER) IS
SELECT salary FROM EMP
WHERE ROWNUM <= n;
BEGIN
-- Open the cursor and loop through the employee records OPEN emp_cur(3);
-- Replace 10 with the desired value of n LOOP
FETCH emp_cur INTO sal_var;
EXIT WHEN emp_cur%NOTFOUND;
-- Add the salary to the total and increment the record count total_sal :=
total_sal + sal_var;
rec_count := rec_count + 1;
END LOOP; CLOSE emp_cur;
-- Display the total salary and record count DBMS_OUTPUT.PUT_LINE('Total
Salary: ' || total_sal);
DBMS_OUTPUT.PUT_LINE('Record Count: ' || rec_count);
END; /
```

```
Statement processed.
Total Salary: 180000
Record Count: 3
```