**Class Test 03**

**PL/SQL**

**Part 01:**

1. Write a query that can multiply two numbers taking input from user.
2. Write a query that can add two numbers if the numbers are equal. Use CASE Statement.
3. Write a query that can check if two strings are equal or not. Use IF-THEN-ELSIF Statement.
4. Write a query that can multiply two numbers. If the result obtained is less than 100, **Hi** is displayed, if the result obtained is more than 100, **Bye** is displayed and if the result obtained is equal to 100, **ADBMS** is displayed. Use IF-THEN-ELSIF Statement
5. Write a query that can check if two numbers are equal or not. Use CASE Statement.

**Part 02:**

*To solve the following use the scott schema*

1. Write a query that can display the salary of employee ALLEN. If ALLEN’s salary is greater than 2000 display ‘SALARY GREATER THAN 2000’ and If not then display ‘SALARY LESS THAN 2000’.
2. Write a query that can ask user to input the EMPNO of employee WARD and display his salary.
3. Write a query that can ask user to input the EMPNO of employee BLAKE,CLARK and TURNER and display their respective salary.
4. Write a query that can ask user to input the EMPNO of employee BLAKE,CLARK and TURNER and display their respective salary, add the salaries and display the total.
5. Write a query that displays the commission of employee SMITH. If SMITH’s commission is NULL. Display ‘NOT APPLICABLE FOR COMMISSION’

**Part 03:**

*To solve the following use the scott schema*

1. Write a query that can display the salary of employee JONES three times using basic loop.
2. Write a query that can display the salary of employee JONES three times using while loop.
3. Write a query that can display the salary of employee JONES three times using for loop.
4. Create a function that returns the total number of departments.
5. Create a procedure to update the salary of employee Allen to 100.

**\*\*After solving the above questions using Oracle 10g, write the PL/SQLs in a MS Word document (Write down the answer and give screenshot of the result of the query. The name of the document MUST be your ID and the PL/SQLs MUST be numbered accordingly) and upload it in the provided link in your VUES account**

1. Write a query that can multiply two numbers taking input from user.

DECLARE

num1 NUMBER := :num1;

num2 NUMBER := :num2;

result NUMBER;

BEGIN

result := num1 \* num2;

DBMS\_OUTPUT.PUT\_LINE('Multiplication Result: ' || result);

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

1. Write a query that can add two numbers if the numbers are equal. Use CASE Statement.

DECLARE

num1 NUMBER := :num1;

num2 NUMBER := :num2;

result NUMBER;

BEGIN

CASE

WHEN num1 = num2 THEN

result := num1 + num2;

DBMS\_OUTPUT.PUT\_LINE('Numbers are equal. Sum = ' || result);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Numbers are not equal. No addition performed.');

END CASE;

END;

/A screenshot of a computer

AI-generated content may be incorrect.

3. Write a query that can check if two strings are equal or not. Use IF-THEN-ELSIF Statement.

DECLARE

str1 VARCHAR2(100) := 'HELLO';

str2 VARCHAR2(100) := 'HELLO';

BEGIN

IF str1 = str2 THEN

DBMS\_OUTPUT.PUT\_LINE('Strings are equal.');

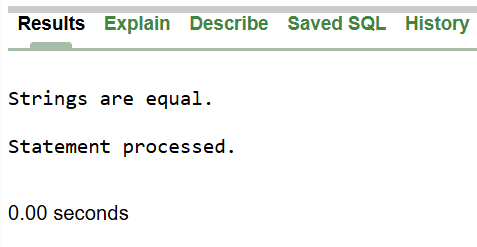
ELSIF str1 != str2 THEN

DBMS\_OUTPUT.PUT\_LINE('Strings are not equal.');

END IF;

END;

/



4. Write a query that can multiply two numbers. If the result obtained is less than 100, Hi is displayed, if the result obtained is more than 100, Bye is displayed and if the result obtained is equal to 100, ADBMS is displayed. Use IF-THEN-ELSIF Statement

DECLARE

num1 NUMBER := 10;

num2 NUMBER := 10;

result NUMBER;

BEGIN

result := num1 \* num2;

IF result < 100 THEN

DBMS\_OUTPUT.PUT\_LINE('Hi');

ELSIF result > 100 THEN

DBMS\_OUTPUT.PUT\_LINE('Bye');

ELSIF result = 100 THEN

DBMS\_OUTPUT.PUT\_LINE('ADBMS');

END IF;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

5. Write a query that can check if two numbers are equal or not. Use CASE Statement.

DECLARE

num1 NUMBER := 100;

num2 NUMBER := 100;

BEGIN

CASE

WHEN num1 = num2 THEN

DBMS\_OUTPUT.PUT\_LINE('Numbers are equal.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Numbers are not equal.');

END CASE;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

6. Write a query that can display the salary of employee ALLEN. If ALLEN’s salary is greater than 2000 display ‘SALARY GREATER THAN 2000’ and If not then display ‘SALARY LESS THAN 2000’.

DECLARE

a\_sal NUMBER;

BEGIN

SELECT sal INTO a\_sal FROM emp WHERE ename = 'ALLEN';

IF a\_sal > 2000 THEN

DBMS\_OUTPUT.PUT\_LINE('SALARY GREATER THAN 2000');

ELSE

DBMS\_OUTPUT.PUT\_LINE('SALARY LESS THAN 2000');

END IF;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

7. Write a query that can ask user to input the EMPNO of employee WARD and display his salary.

DECLARE

w\_empno NUMBER;

w\_sal NUMBER;

BEGIN

w\_empno := :w\_empno;

SELECT sal INTO w\_sal FROM emp WHERE empno = w\_empno AND ename = 'WARD';

DBMS\_OUTPUT.PUT\_LINE('Salary of WARD: ' || w\_sal);

END;

/

|  |  |
| --- | --- |
|  |  |

8. Write a query that can ask user to input the EMPNO of employee BLAKE, CLARK and TURNER and display their respective salary.

DECLARE

empno\_blake NUMBER;

empno\_clark NUMBER;

empno\_turner NUMBER;

salary\_blake NUMBER;

salary\_clark NUMBER;

salary\_turner NUMBER;

BEGIN

empno\_blake := :empno\_blake;

empno\_clark := :empno\_clark;

empno\_turner := :empno\_turner;

SELECT sal INTO salary\_blake FROM emp WHERE empno = empno\_blake AND ename = 'BLAKE';

SELECT sal INTO salary\_clark FROM emp WHERE empno = empno\_clark AND ename = 'CLARK';

SELECT sal INTO salary\_turner FROM emp WHERE empno = empno\_turner AND ename = 'TURNER';

DBMS\_OUTPUT.PUT\_LINE('Salary of BLAKE: ' || salary\_blake);

DBMS\_OUTPUT.PUT\_LINE('Salary of CLARK: ' || salary\_clark);

DBMS\_OUTPUT.PUT\_LINE('Salary of TURNER: ' || salary\_turner);

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

9. Write a query that can ask user to input the EMPNO of employee BLAKE, CLARK and TURNER and display their respective salary, add the salaries and display the total.

DECLARE

empno\_blake NUMBER;

empno\_clark NUMBER;

empno\_turner NUMBER;

salary\_blake NUMBER;

salary\_clark NUMBER;

salary\_turner NUMBER;

total\_salary NUMBER;

BEGIN

empno\_blake := :empno\_blake;

empno\_clark := :empno\_clark;

empno\_turner := :empno\_turner;

SELECT sal INTO salary\_blake FROM emp WHERE empno = empno\_blake AND ename = 'BLAKE';

SELECT sal INTO salary\_clark FROM emp WHERE empno = empno\_clark AND ename = 'CLARK';

SELECT sal INTO salary\_turner FROM emp WHERE empno = empno\_turner AND ename = 'TURNER';

total\_salary := salary\_blake + salary\_clark + salary\_turner;

DBMS\_OUTPUT.PUT\_LINE('Salary of BLAKE: ' || salary\_blake);

DBMS\_OUTPUT.PUT\_LINE('Salary of CLARK: ' || salary\_clark);

DBMS\_OUTPUT.PUT\_LINE('Salary of TURNER: ' || salary\_turner);

DBMS\_OUTPUT.PUT\_LINE('Total Salary: ' || total\_salary);

END;

/

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. | A screenshot of a computer  AI-generated content may be incorrect. |

10. Write a query that displays the commission of employee SMITH. If SMITH’s commission is NULL. Display ‘NOT APPLICABLE FOR COMMISSION’

DECLARE

s\_comm NUMBER;

BEGIN

SELECT COMM INTO s\_comm FROM emp WHERE ename = 'SMITH';

IF s\_comm IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('NOT APPLICABLE FOR COMMISSION');

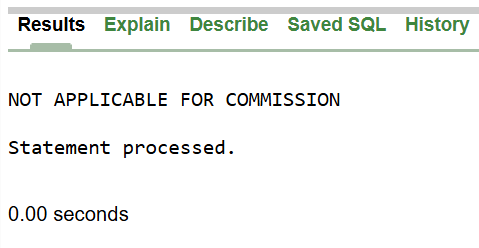
ELSE

DBMS\_OUTPUT.PUT\_LINE('Commission: ' || s\_comm);

END IF;

END;

/



11. Write a query that can display the salary of employee JONES three times using basic loop.

DECLARE

j\_salary NUMBER;

counter NUMBER := 1;

BEGIN

SELECT sal INTO j\_salary FROM emp WHERE ename = 'JONES';

LOOP

DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || j\_salary);

counter := counter + 1;

IF counter > 3 THEN

EXIT;

END IF;

END LOOP;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

12. Write a query that can display the salary of employee JONES three times using while loop.

DECLARE

j\_salary NUMBER;

counter NUMBER := 1;

BEGIN

SELECT sal INTO j\_salary FROM emp WHERE ename = 'JONES';

WHILE counter <= 3 LOOP

DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || j\_salary);

counter := counter + 1;

END LOOP;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

13. Write a query that can display the salary of employee JONES three times using for loop.

DECLARE

j\_salary NUMBER;

BEGIN

SELECT sal INTO j\_salary FROM emp WHERE ename = 'JONES';

FOR i IN 1..3 LOOP

DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || j\_salary);

END LOOP;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

14. Create a function that returns the total number of departments.

CREATE OR REPLACE FUNCTION get\_total\_depts

RETURN NUMBER IS

total\_depts NUMBER;

BEGIN

SELECT COUNT(\*) INTO total\_depts FROM dept;

RETURN total\_depts;

END;

/

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect. |  |

15. Create a procedure to update the salary of employee Allen to 100.

CREATE OR REPLACE PROCEDURE update\_allen\_salary IS

BEGIN

UPDATE emp SET sal = 100

WHERE ename = 'ALLEN';

END;

/

A screenshot of a computer

AI-generated content may be incorrect.