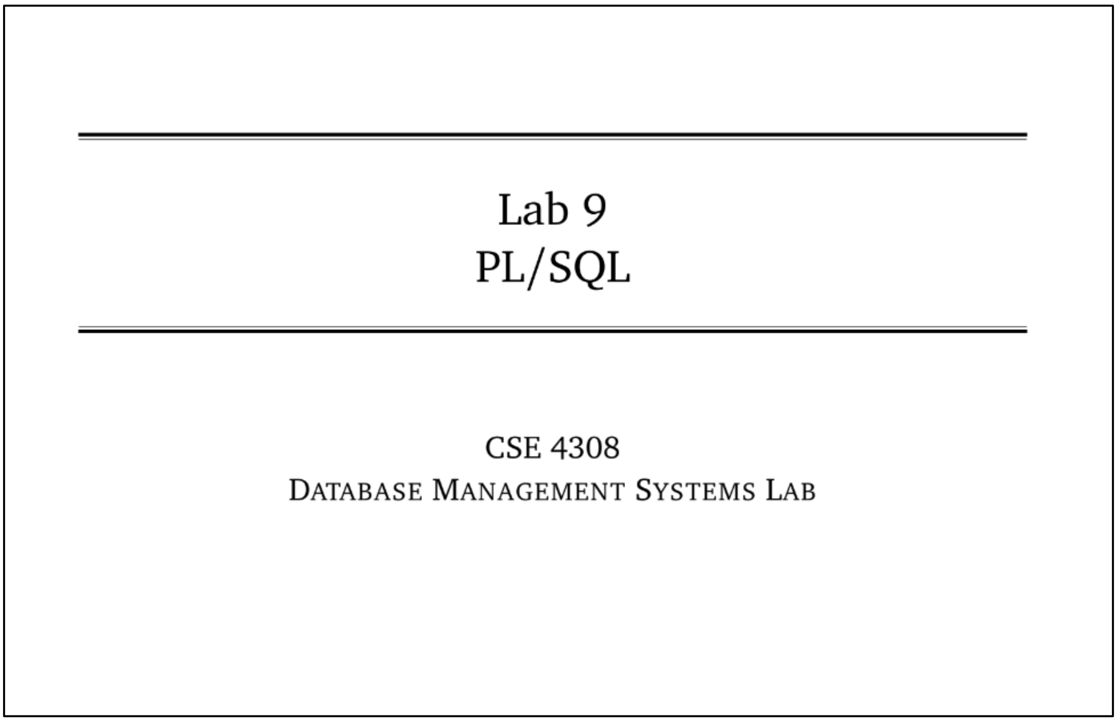
LAB REPORT

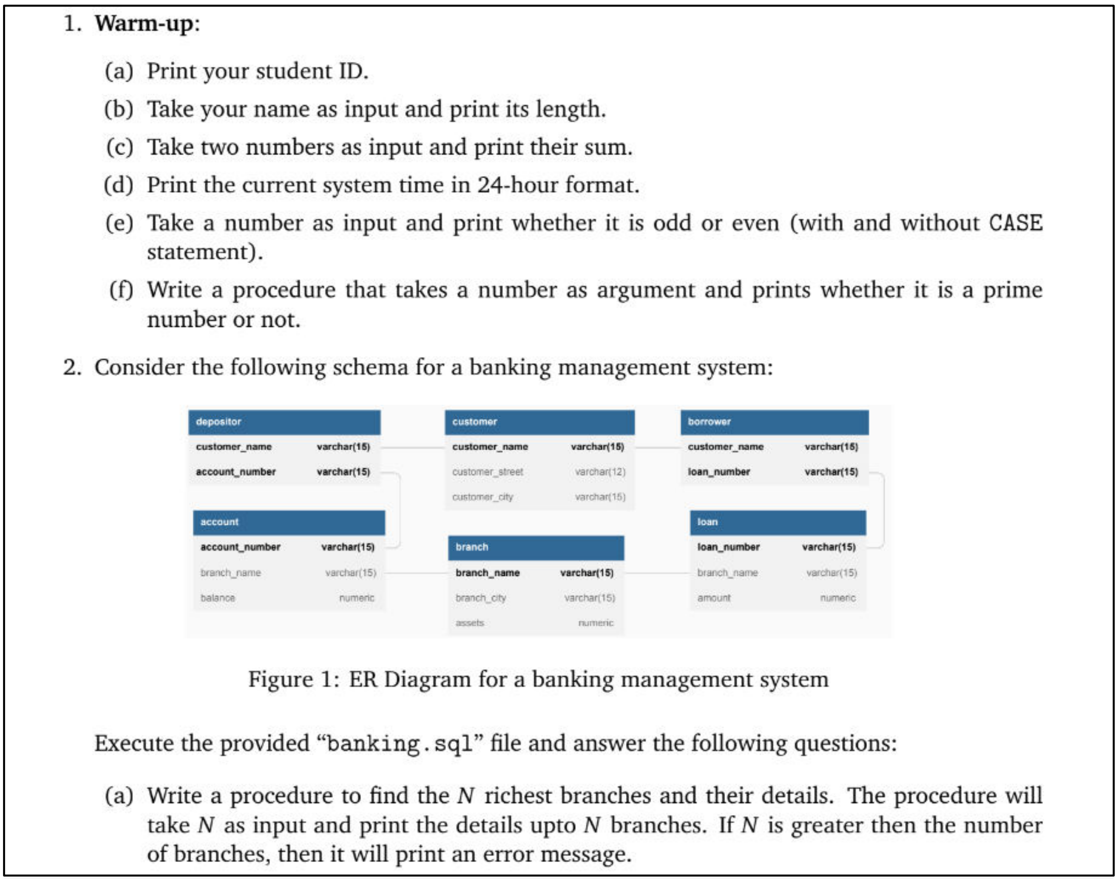


Lab 9
PL/SQL

CSE 4308
DATABASE MANAGEMENT SYSTEMS LAB

NAME: CHOWDHURY ASHFAQ
STUDENT ID: 200042123
PROGRAM: SWE
GROUP: 1A
DATE: 06/11/22

Tasks:

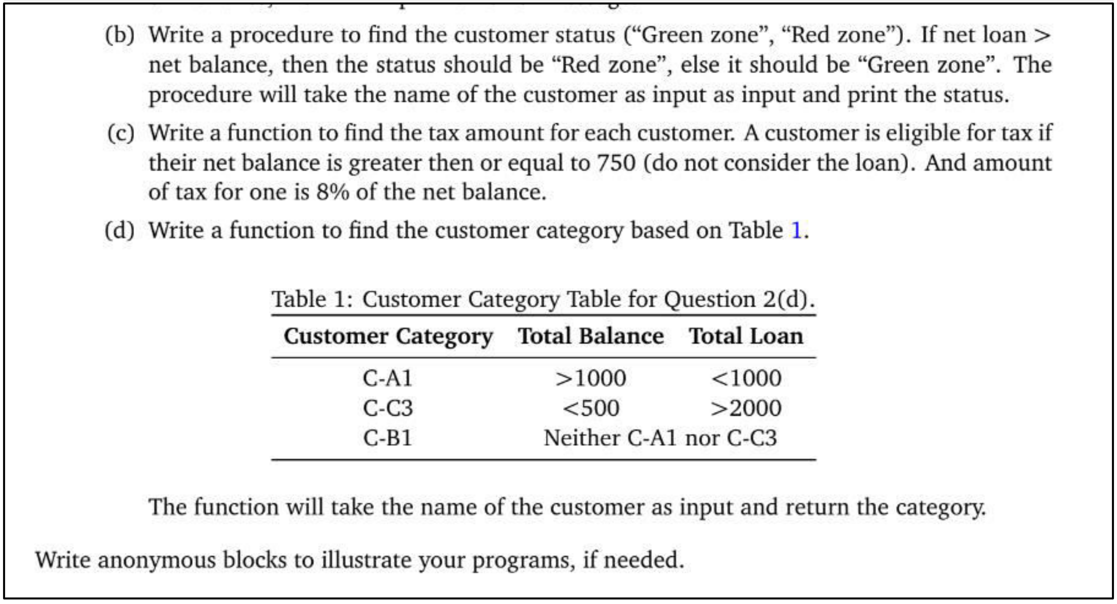


1. **Warm-up**:
   - (a) Print your student ID.
   - (b) Take your name as input and print its length.
   - (c) Take two numbers as input and print their sum.
   - (d) Print the current system time in 24-hour format.
   - (e) Take a number as input and print whether it is odd or even (with and without `CASE` statement).
   - (f) Write a procedure that takes a number as argument and prints whether it is a prime number or not.
2. Consider the following schema for a banking management system:

Figure 1: ER Diagram for a banking management system

Execute the provided “`banking.sql`” file and answer the following questions:

(a) Write a procedure to find the *N* richest branches and their details. The procedure will take *N* as input and print the details upto *N* branches. If *N* is greater then the number of branches, then it will print an error message.



(b) Write a procedure to find the customer status (“Green zone”, “Red zone”). If net loan > net balance, then the status should be “Red zone”, else it should be “Green zone”. The procedure will take the name of the customer as input as input and print the status.

(c) Write a function to find the tax amount for each customer. A customer is eligible for tax if their net balance is greater then or equal to 750 (do not consider the loan). And amount of tax for one is 8% of the net balance.

(d) Write a function to find the customer category based on Table 1.

Table 1: Customer Category Table for Question 2(d).

Customer Category	Total Balance	Total Loan
C-A1	>1000	<1000
C-C3	<500	>2000
C-B1	Neither C-A1 nor C-C3	

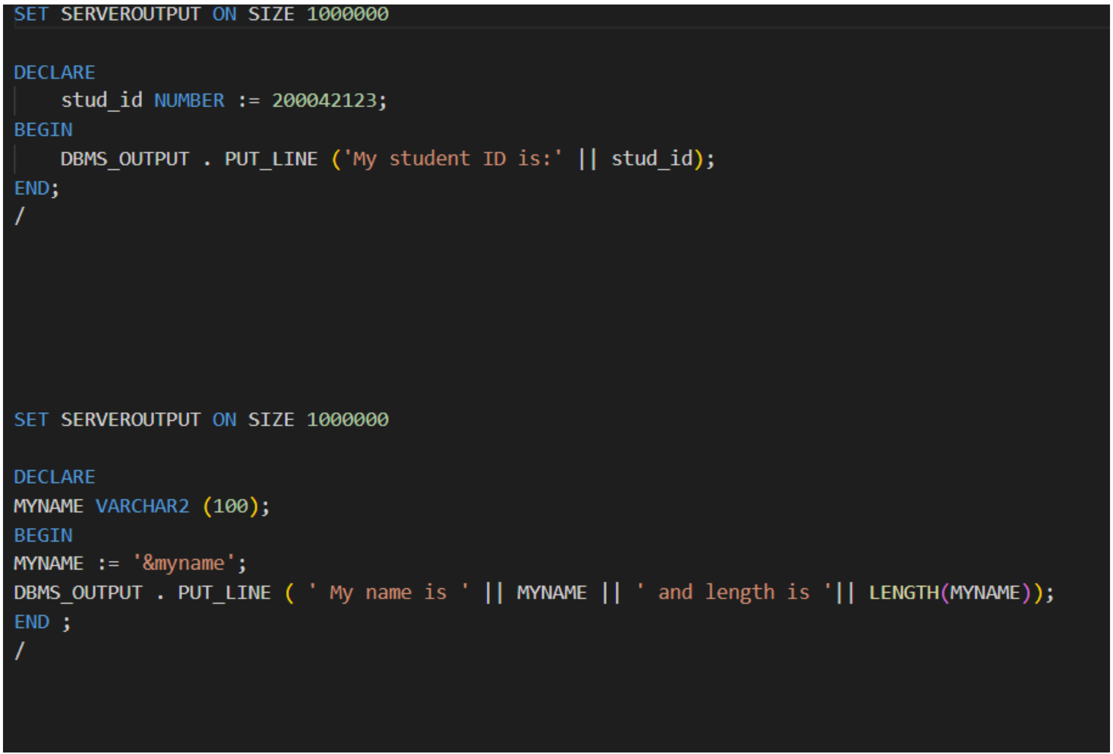
The function will take the name of the customer as input and return the category.

Write anonymous blocks to illustrate your programs, if needed.

Analysis of the problem:

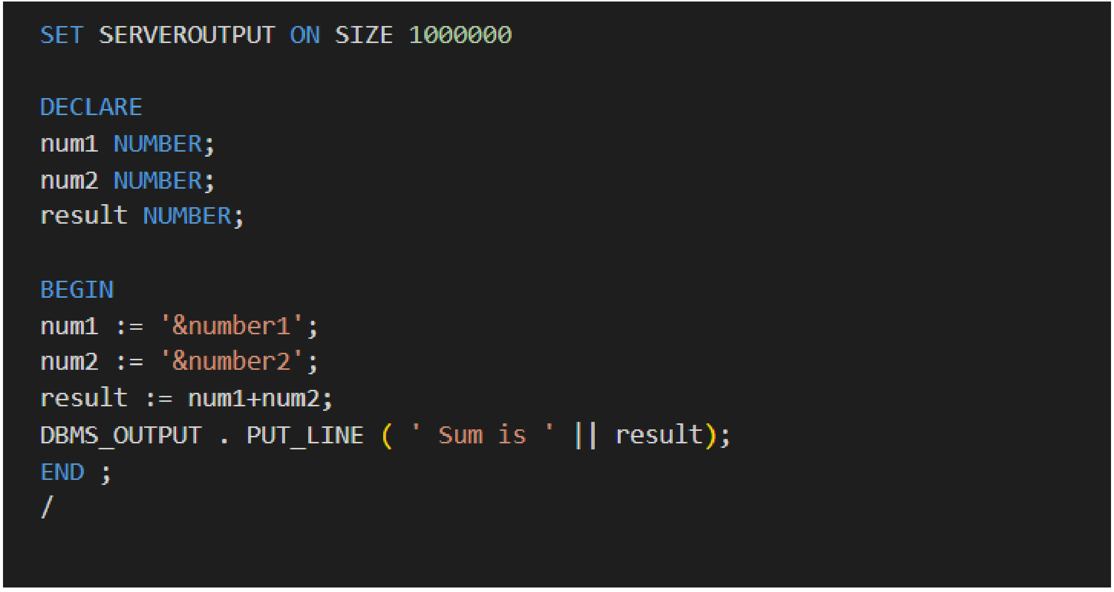
At first we needed to write some very basic statements using PL/SQL. Later we had to write some functions and procedures to perform some particular tasks mentioned above.

Solution:

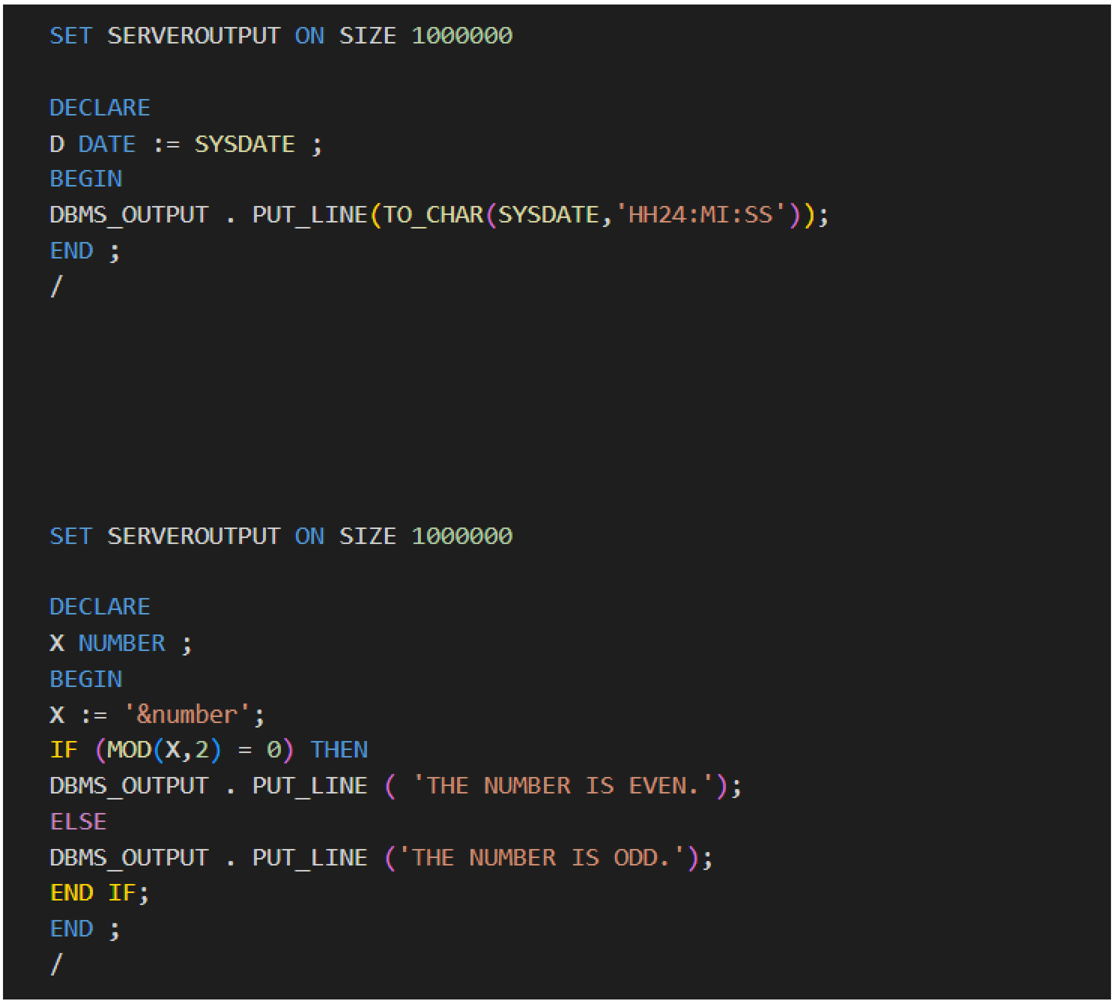


```
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
    stud_id NUMBER := 200042123;
BEGIN
    DBMS_OUTPUT . PUT_LINE ('My student ID is:' || stud_id);
END;
/
```

```
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
MYNAME VARCHAR2 (100);
BEGIN
MYNAME := '&myname';
DBMS_OUTPUT . PUT_LINE ( ' My name is ' || MYNAME || ' and length is '|| LENGTH(MYNAME));
END ;
/
```

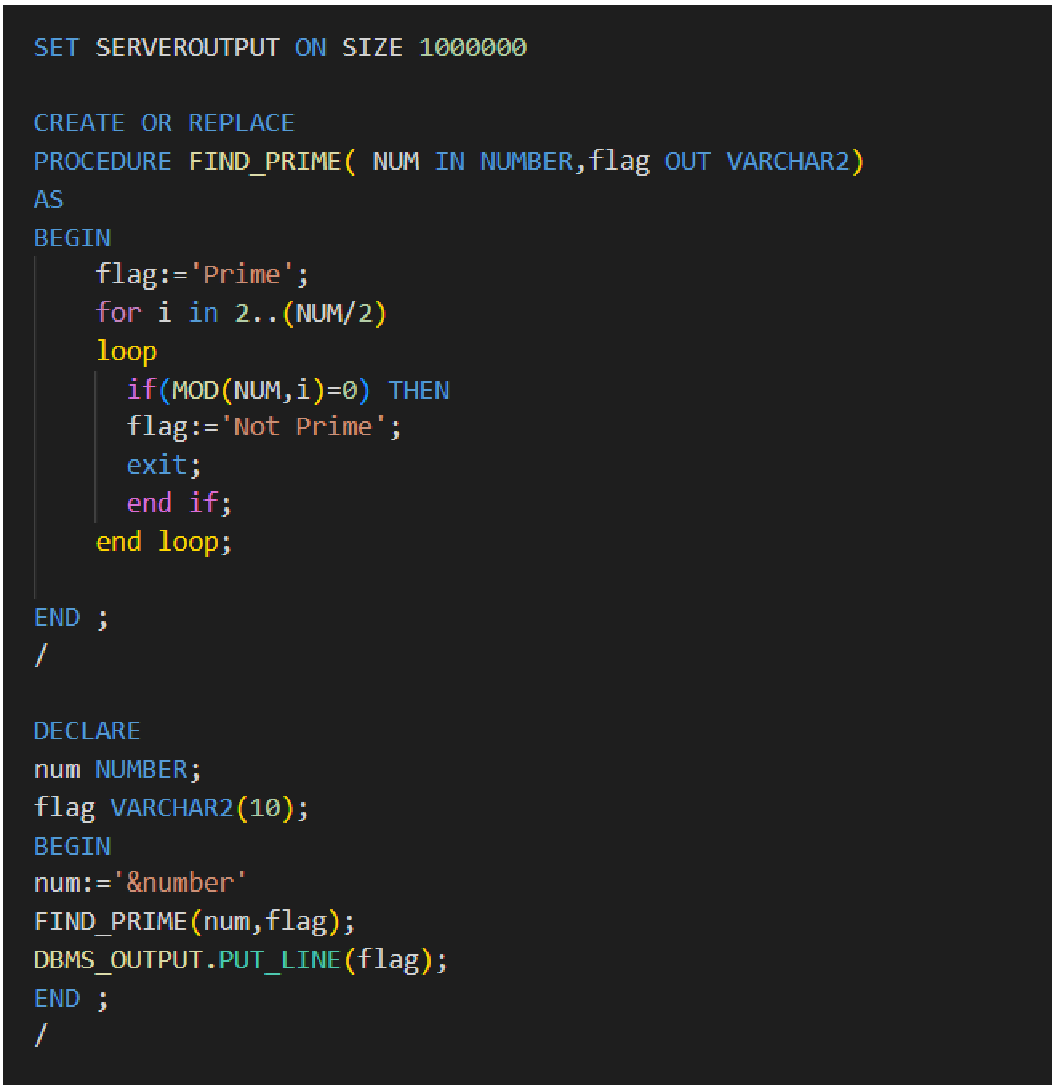


```
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
num1 NUMBER;
num2 NUMBER;
result NUMBER;

BEGIN
num1 := '&number1';
num2 := '&number2';
result := num1+num2;
DBMS_OUTPUT . PUT_LINE ( ' Sum is ' || result);
END ;
/
```

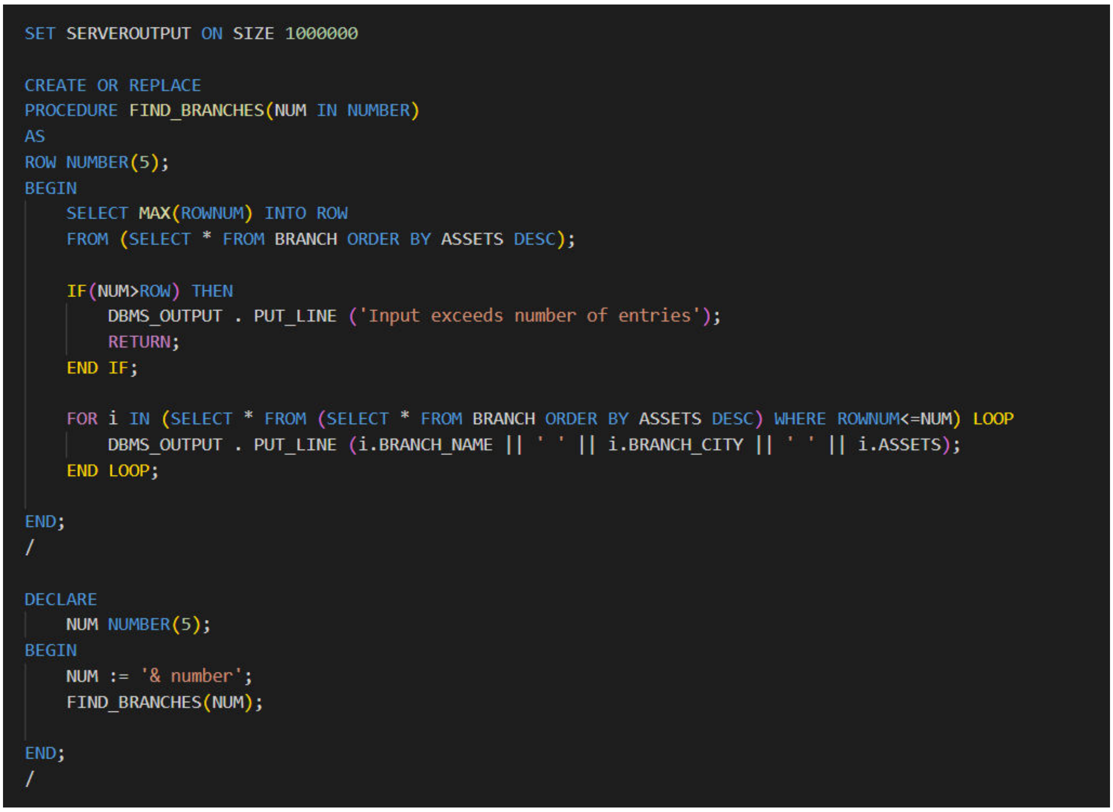


```
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
D DATE := SYSDATE ;
BEGIN
DBMS_OUTPUT . PUT_LINE(TO_CHAR(SYSDATE,'HH24:MI:SS'));
END ;
/
```

```
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
X NUMBER ;
BEGIN
X := '&number';
IF (MOD(X,2) = 0) THEN
DBMS_OUTPUT . PUT_LINE ( 'THE NUMBER IS EVEN.');
ELSE
DBMS_OUTPUT . PUT_LINE ('THE NUMBER IS ODD.');
END IF;
END ;
/
```



```
SET SERVEROUTPUT ON SIZE 1000000

CREATE OR REPLACE
PROCEDURE FIND_PRIME( NUM IN NUMBER,flag OUT VARCHAR2)
AS
BEGIN
    flag:='Prime';
    for i in 2..(NUM/2)
    loop
      if(MOD(NUM,i)=0) THEN
      flag:='Not Prime';
      exit;
      end if;
    end loop;

END ;
/

DECLARE
num NUMBER;
flag VARCHAR2(10);
BEGIN
num:='&number'
FIND_PRIME(num,flag);
DBMS_OUTPUT.PUT_LINE(flag);
END ;
/
```



```
SET SERVEROUTPUT ON SIZE 1000000

CREATE OR REPLACE
PROCEDURE FIND_BRANCHES(NUM IN NUMBER)
AS
ROW NUMBER(5);
BEGIN
    SELECT MAX(ROWNUM) INTO ROW
    FROM (SELECT * FROM BRANCH ORDER BY ASSETS DESC);

    IF(NUM>ROW) THEN
        DBMS_OUTPUT . PUT_LINE ('Input exceeds number of entries');
        RETURN;
    END IF;

    FOR i IN (SELECT * FROM (SELECT * FROM BRANCH ORDER BY ASSETS DESC) WHERE ROWNUM<=NUM) LOOP
        DBMS_OUTPUT . PUT_LINE (i.BRANCH_NAME || ' ' || i.BRANCH_CITY || ' ' || i.ASSETS);
    END LOOP;

END;
/

DECLARE
    NUM NUMBER(5);
BEGIN
    NUM := '& number';
    FIND_BRANCHES(NUM);

END;
/
```