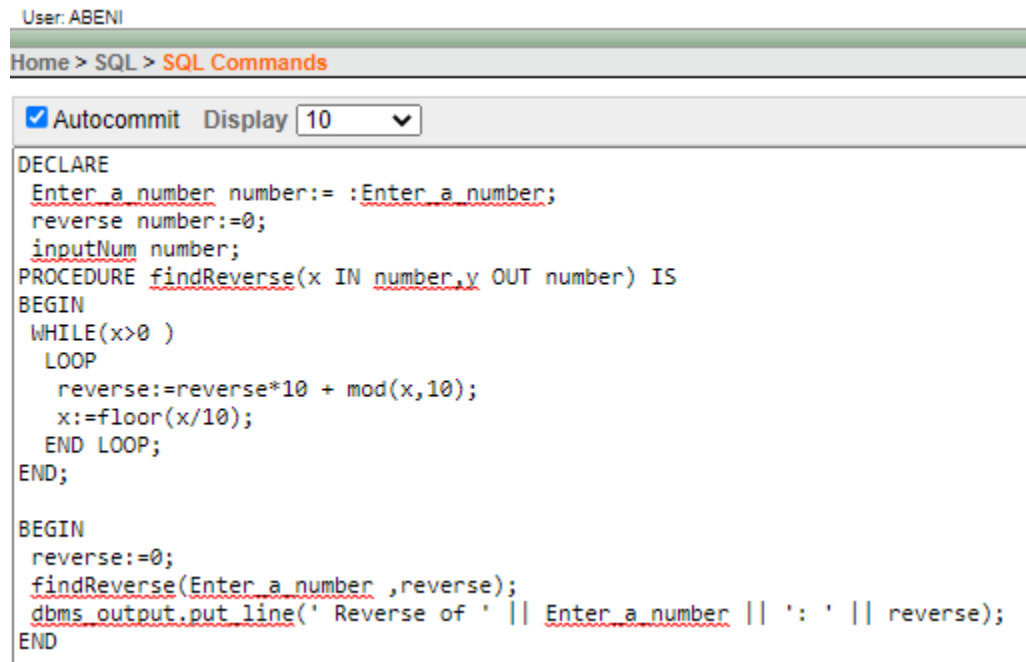


1. Write PL/SQL code in Procedure to find Reverse number



The screenshot shows a SQL command window with the following content:

```
User: ABENI
Home > SQL > SQL Commands

Autocommit Display 10

DECLARE
  Enter a number number:= :Enter a number;
  reverse number:=0;
  inputNum number;
PROCEDURE findReverse(x IN number,y OUT number) IS
BEGIN
  WHILE(x>0 )
  LOOP
    reverse:=reverse*10 + mod(x,10);
    x:=floor(x/10);
  END LOOP;
END;

BEGIN
  reverse:=0;
  findReverse(Enter a number ,reverse);
  dbms_output.put_line(' Reverse of ' || Enter a number || ': ' || reverse);
END
```

Basically, we just created the procedure and added the main code in it

The procedure handles the reverse process

To find the reverse, we initialized the reversed value as 0 then added the initial reversed number and the module of our input by 10 until our input reaches the false state of our while loop

We called the procedure in another block

Printed out a message of the reversed number

2. Write PL/SQL code in Procedure to find Factorial of a given number by using call procedure and the procedure should make use recursive function

```
User: ABENI
Home > SQL > SQL Commands
Autocommit Display 10
DECLARE
  num number := :num;
  factorial number := 1;
  x number;
  msg char;
PROCEDURE findFactorial(x IN OUT number, y OUT char) IS
BEGIN
  WHILE( x>0 )
  LOOP
    factorial := factorial*x;
    x:= x-1;
  END LOOP;
  y:=factorial;
END;

BEGIN
  findFactorial(num,msg);
  dbms_output.put_line('factorial of ' || num || ' is ' || msg);
END;
```

Created a procedure and then added the main function in it

We initialized our final factorial result as 1

Our main code in the while loop assigns the multiple of the initial factorial value with our input with every loop deducting it by

When our input value makes the while loop false, we return the final value by calling the procedure

We called the procedure in another block

3. Write a Procedure to check the given number is prime or not by using call procedure.

```
User: ABENI
Home > SQL > SQL Commands

Autocommit Display 10

DECLARE
  Enter_a_number number := :Enter_a_number;
  counter number := 0;
  x number;
  i number;
  msg char;
PROCEDURE primeOrNot(x IN OUT number, y OUT char) IS
BEGIN
  i:=x;
  WHILE( x>0 )
  LOOP
    IF((mod(x,i))=0)
    THEN
      counter:=counter+1;
    END IF;
    i:= i-1;
  END LOOP;
  IF(counter>2)
  THEN
    y:=(Enter_a_number|| ' is not a prime number');
  ELSE
    y:=(Enter_a_number|| ' is a prime number');
  END IF;
END;

BEGIN
  primeOrNot(Enter_a_number,msg);
  dbms_output.put_line(msg);
END;
```

we used a call procedure to check if a number is prime or not
It gets an input number and keeps a counter to check if the number is divisible by one of the numbers in the while loop.
If the counter is greater than 2 then we decide that the number is not prime
We called the procedure in another block

4. Write a Function to check the given number is prime or not.

```
User: ABENI
Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

DECLARE
  Enter_a_number number := :Enter_a_number;
  counter number := 0;
  x number;
  i number;
  msg char;
FUNCTION primeOrNot(x IN OUT number) RETURN number IS y char
BEGIN
  i:=x;
  WHILE( x>0 )
  LOOP
    IF((mod(x,i))=0)
    THEN
      counter:=counter+1;
    END IF;
    i:= i-1;
  END LOOP;
  IF(counter>2)
  THEN
    y:=(Enter_a_number|| ' is not a prime number');
  ELSE
    y:=(Enter_a_number|| ' is a prime number');
  END IF;
  return y;
END;

BEGIN
  msg:=primeOrNot(Enter_a_number);
  dbms_output.put_line(msg);
END;
```

Same as the above question but instead of procedure we used function

Here we returned the message in the function we created above

We not only called the function in another block but we assigned the returned value of the function to a variable and printed that value

5. Describe what the main difference between Procedure and Function in PL/SQL

*When using procedures, it's not mandatory to return a value, while its quite the opposite in functions.

*we mainly use procedures to execute process while we use functions to perform calculations.