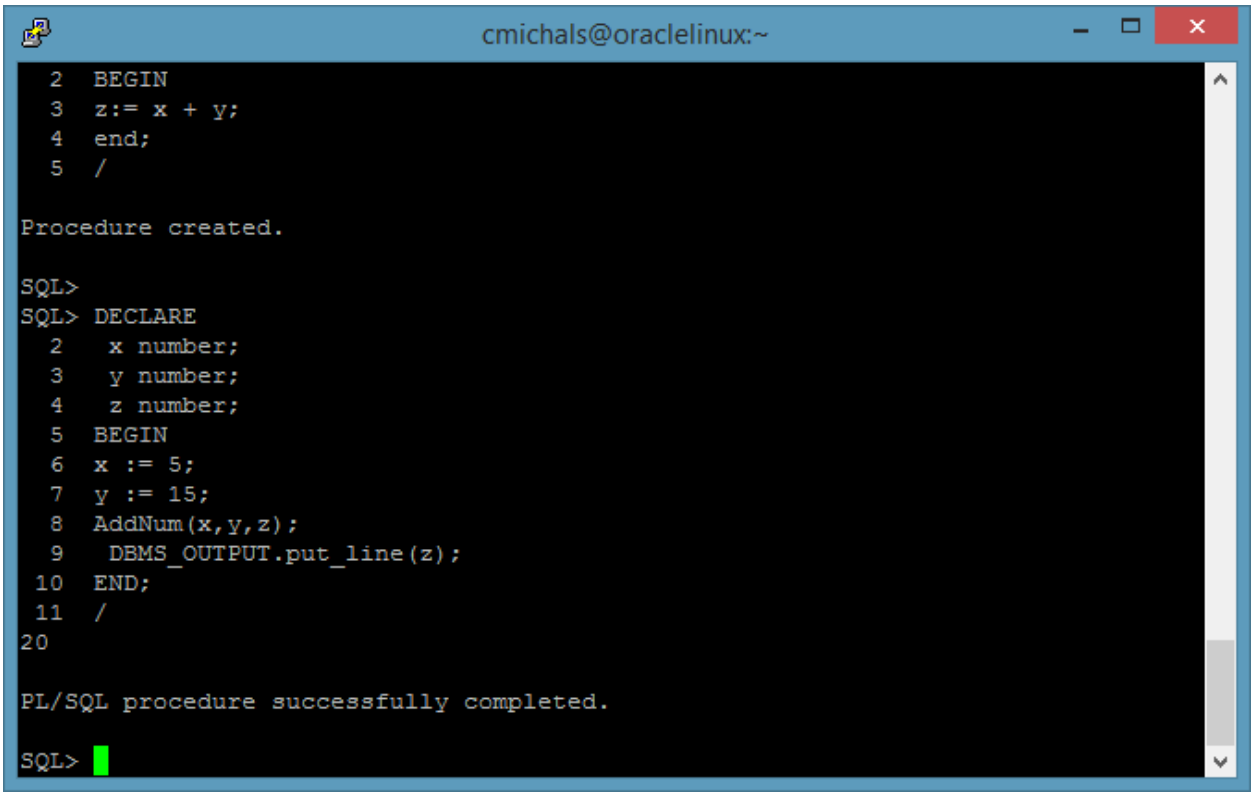
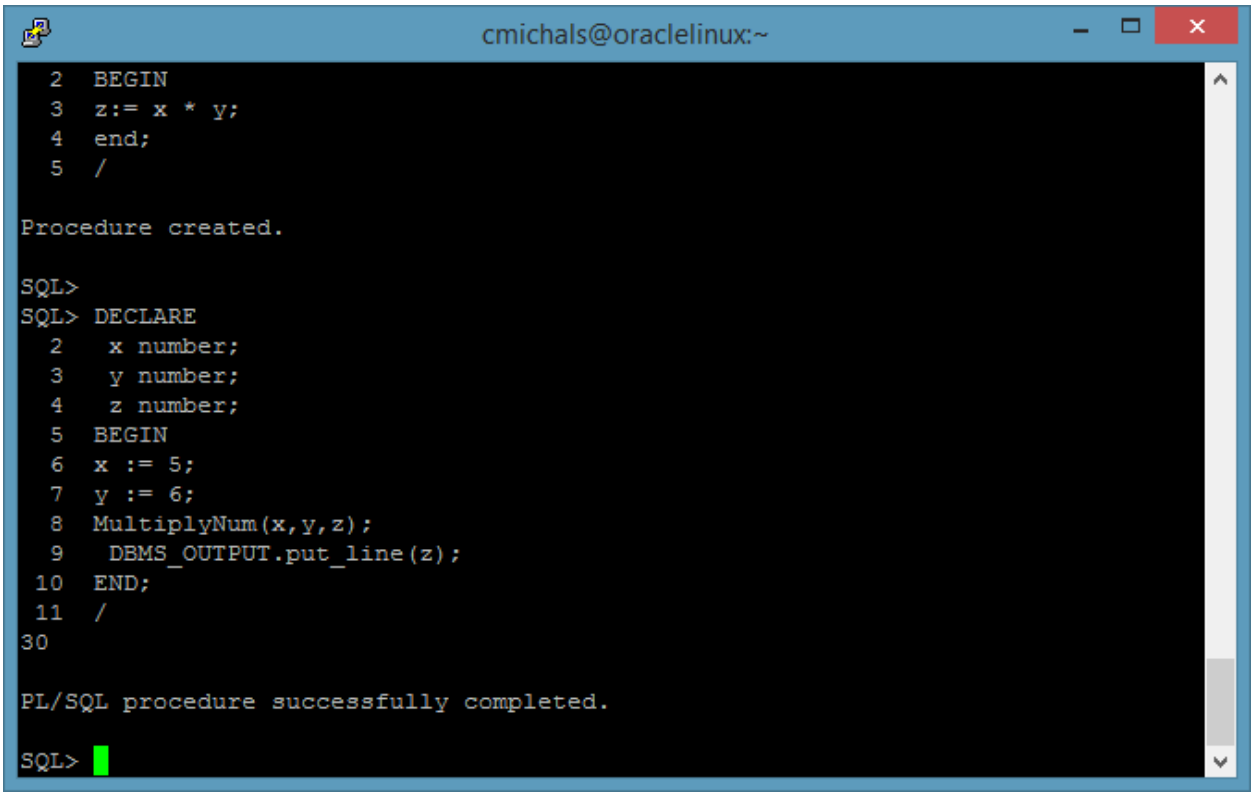
-- 1: Proc AddNum: this proc will add two numbers X & Y, and return the result in a third variable Z

```
create or REPLACE PROCEDURE AddNum(x IN number, y IN number, z OUT number)
AS
BEGIN
z:= x + y;
end;
/

DECLARE
 x number;
 y number;
 z number;
BEGIN
x := 5;
y := 15;
AddNum(x,y,z);
 DBMS_OUTPUT.put_line(z);
END;
/
```



-- 2: Proc MultiplyNum: this proc will multiply two numbers X & Y, and return the result in a third variable Z

```
create or REPLACE PROCEDURE MultiplyNum(x IN number, y IN number, z OUT number) AS
BEGIN
z:= x * y;
end;
/

DECLARE
 x number;
 y number;
 z number;
BEGIN
x := 5;
y := 6;
MultiplyNum(x,y,z);
 DBMS_OUTPUT.put_line(z);
END;
/
```



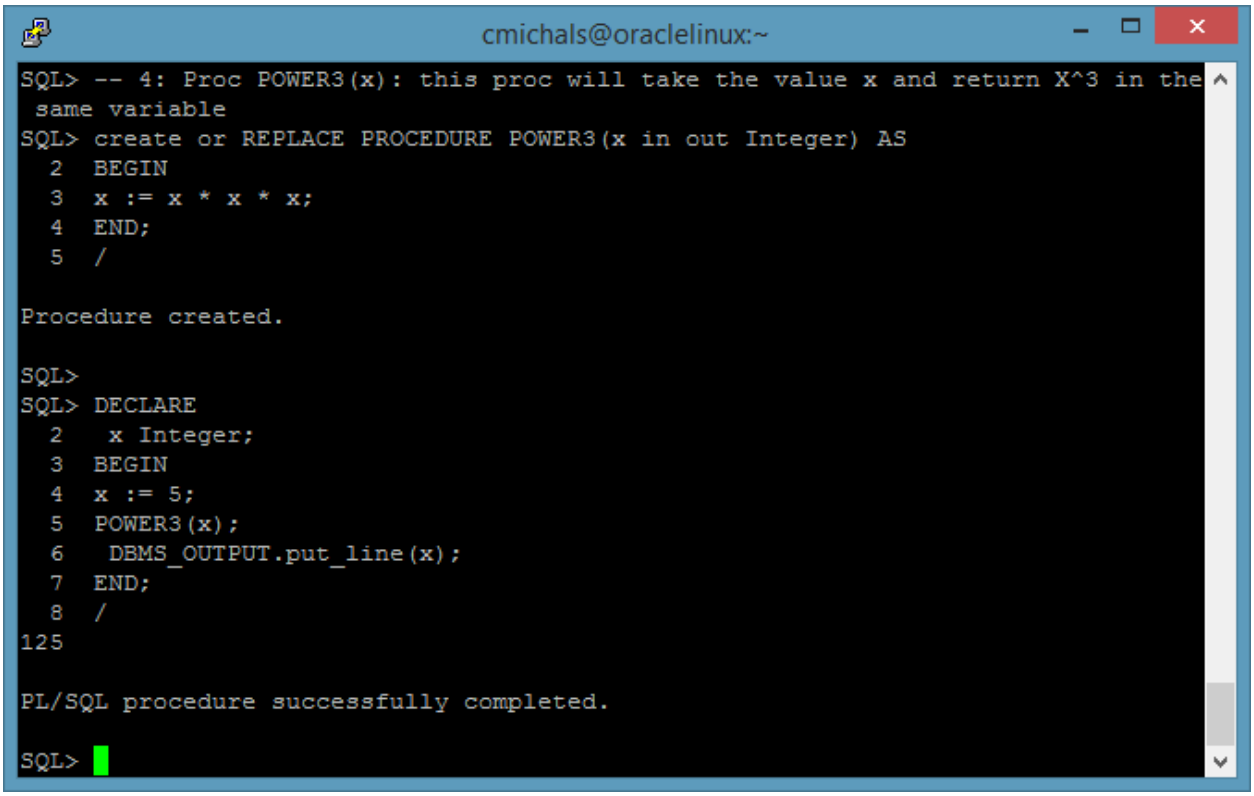
```
-- 3: Porc POWERR(n,r, R): this proc will raise the first number n, to the power r &
return the result in the variable R.
create or REPLACE PROCEDURE POWERR(n IN Integer, r IN Integer, O OUT Integer)
AS
loopCounter Integer;
BEGIN
  loopCounter := 0;
        IF r < 0 THEN
                raise_application_error(-20101, 'Can not calculate negative exponents');
        END IF;
        if r = 0 THEN
                O := 0;
                return;
        END IF;
        O := n;
        loop
                exit when loopCounter = r;
                O := n * O;
                loopCounter := loopCounter + 1;
        end loop;
end;
/

DECLARE
 x Integer;
 y Integer;
 z Integer;
BEGIN
x := 5;
y := 6;
POWERR(x,y,z);
 DBMS_OUTPUT.put_line(z);
END;
/
```

cmichals@oraclelinux:~

```
 16                 loopCounter := loopCounter + 1;
 17          end loop;
 18  end;
 19  /

Procedure created.

SQL>
SQL> DECLARE
  2   x Integer;
  3   y Integer;
  4   z Integer;
  5  BEGIN
  6  x := 5;
  7  y := 6;
  8  POWERR(x,y,z);
  9   DBMS_OUTPUT.put_line(z);
 10  END;
 11  /
78125

PL/SQL procedure successfully completed.

SQL>
```

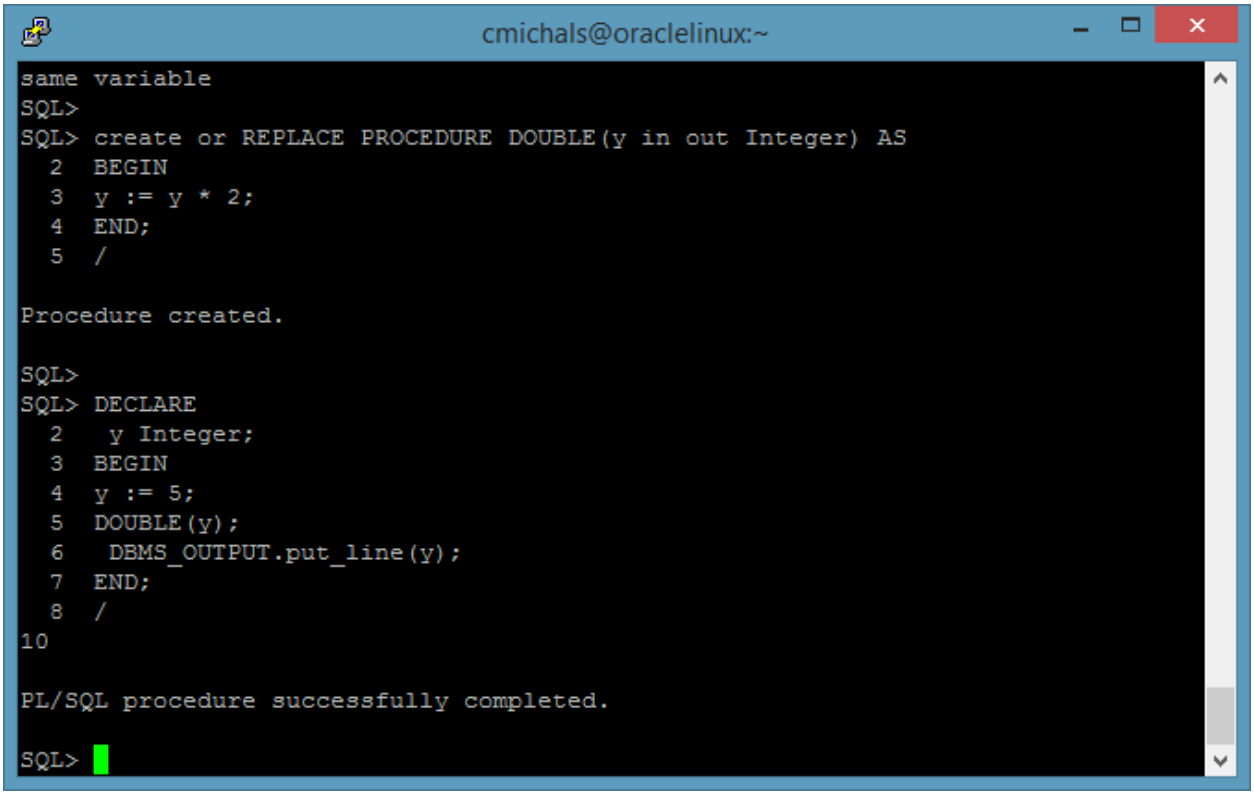
-- 4: Proc POWER3(x): this proc will take the value x and return X^3 in the same variable

```
create or REPLACE PROCEDURE POWER3(x in out Integer) AS
BEGIN
x := x * x * x;
END;
/

DECLARE
 x Integer;
BEGIN
x := 5;
POWER3(x);
 DBMS_OUTPUT.put_line(x);
END;
/
```



-- 5: Proc DOUBLE(y): this proc will take the value y and return 2y in the same variable

```
create or REPLACE PROCEDURE DOUBLE(y in out Integer) AS
BEGIN
y := y * 2;
END;
/

DECLARE
 y Integer;
BEGIN
y := 5;
DOUBLE(y);
 DBMS_OUTPUT.put_line(y);
END;
/
```



-- 6: Proc printHi(first, last): this proc takes two strings, first name & last name, and prints: “Hi <first last> to the
-- console. E.g. calling printHi(‘Blake’, ‘Griffin’) , results in: Hi Blake Griffin being printed out.

```
create or REPLACE PROCEDURE printHi(first in VarChar2, last in VarChar2) AS
BEGIN
DBMS_OUTPUT.put_line('Hi '||first||' '||last);
END;
/

DECLARE
 x varchar2(200);
 y varchar2(200);
 BEGIN
x := 'Chet';
y := 'Michals';
printHi(x,y);
end;
/
```