PLSQ PROCEDURE & FUNCTION

BY- N K SHUKLA

FUNCTION

- ❑The function program has a block of code that performs some specific tasks or functions.
- ❑The function can be either user-defined or predefined.

SYNTAX:

```
CREATE [OR REPLACE] FUNCTION function_name [parameters]
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}

BEGIN
   < function_body >
END [function_name];
```

FUNCTION

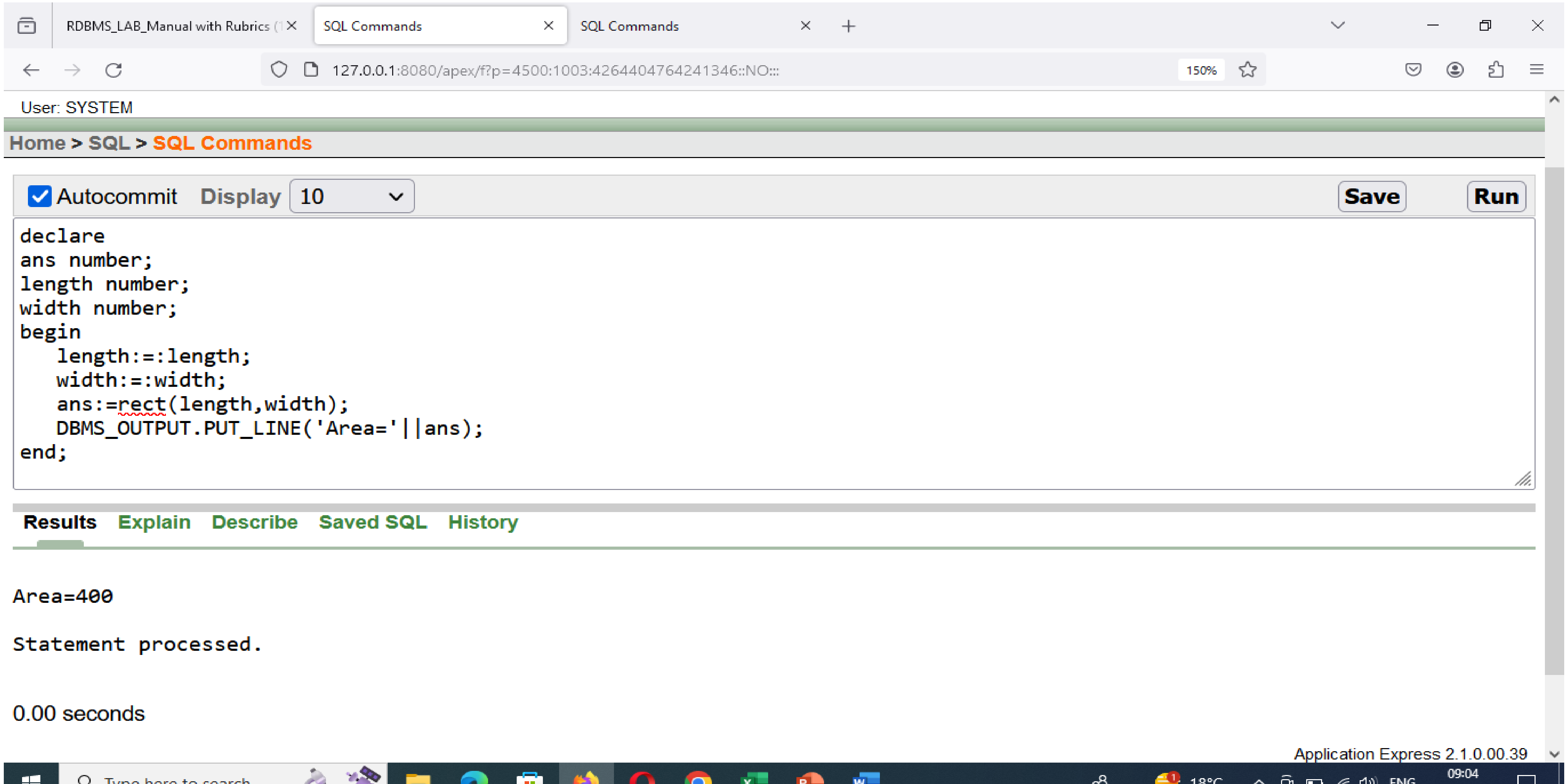
- ❑**IN** represents that value will be passed from outside and OUT represents that this parameter will be used to return a value outside of the procedure.
- ❑RETURN clause specifies that data type you are going to return from the function.
- ❑The AS keyword is used instead of the IS keyword for creating a standalone function.
- ❑EXAMPLE -1 : Create function for multiplication of two integer digit:

Function BODY:

```
create or replace function rect(l in number,w in number) return
number is
begin
 return(l*w);
end;
```

FUNCTION

❑Call a function and output::


RDBMS_LAB_Manual with Rubrics
SQL Commands
SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:4264404764241346::NO:::
150%
User: SYSTEM
Home > SQL > SQL Commands
Autocommit
Display
10
Save
Run
Results
Explain
Describe
Saved SQL
History
Application Express 2.1.0.00.39

```
declare
ans number;
length number;
width number;
begin
   length:=:length;
   width:=:width;
   ans:=rect(length,width);
   DBMS_OUTPUT.PUT_LINE('Area='||ans);
end;
```

```
Area=400

Statement processed.


0.00 seconds
```

PROCEDURE

❑he PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks.

❑The procedure contains a header and a body.

•**Header:** The header contains the name of the procedure and the parameters or variables passed to the procedure.

•**Body:** The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.

❑How to pass parameters in procedure:

1.**IN parameters:** The IN parameter can be referenced by the procedure or function. The value of the parameter cannot be overwritten by the procedure or the function.

2.**OUT parameters:** The OUT parameter cannot be referenced by the procedure or function, but the value of the parameter can be overwritten by the procedure or function.

3.**INOUT parameters:** The INOUT parameter can be referenced by the procedure or function and the value of the parameter can be overwritten by the procedure or function.

PROCEDURE

SYNTAX:

```
CREATE [OR REPLACE] PROCEDURE procedure_name
    [ (parameter [,parameter]) ]
IS
    [declaration_section]
BEGIN
    executable_section
[EXCEPTION
    exception_section]
END [procedure_name];
```

PROCEDURE

PROGRAM 1: calculate the perimeter of a rectangle.
Procedure code:

```
create or replace procedure RECT_PERI(l in number,w in number) is
peri number(8);

begin
    peri:=2*(l+w);
    DBMS_OUTPUT.PUT_LINE(peri);
end;
```

Procedure:

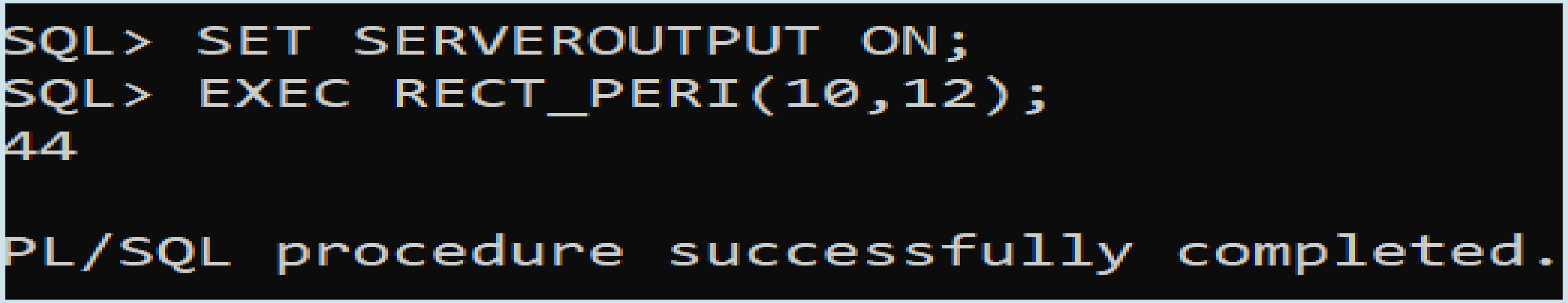
Call RECT_PERI () Procedure from PL/SQL block:

```
DECLARE
length number(8);
width number(8);

BEGIN
length:=:length;
width:=:width;
RECT_PERI(length,width);

END;
```

Call RECT_PERI () Procedure from command line



```
SQL> SET SERVEROUTPUT ON;
SQL> EXEC RECT_PERI(10,12);
44

PL/SQL procedure successfully completed.
```

PROCEDURE

PROGRAM 2 : Write a procedure to check whether given number is odd or even. Also write the PL/SQL block to invoke it.

```
create or replace procedure odd_even(n in number) is

BEGIN
 if n mod 2 = 0 then
    DBMS_OUTPUT.PUT_LINE('NO IS EVEN');
 else
    DBMS_OUTPUT.PUT_LINE('NO IS ODD');
 end if;
END;
```

Procedure:

Call odd_even () Procedure from PL/SQL block:

```
DECLARE
        no number(8);

BEGIN
       no:=:no;
      odd_even(no);

END;
```