12
Demonstrate Bind and Substitution Variables using PL SQL block

Bind Variables

- Oracle provides the bind variable functionality to the user, where we can create the variable in a host environment and use that variable in different PL/SQL blocks as per our requirement.
- We can create bind variables with different data types.

Bind Variables

Declaration of bind variable:

Syntax:

VARIABLE variable_name **DATA TYPE**;

- **Example**: VARIABLE marks NUMBER;

Bind Variables

Referencing in a PL/SQL block

- Bind variables declared in the host environment can be used in a PL/SQL block prefixed with colon symbol (**:**)
- **For example**: Initializing variable **marks**

```
BEGIN
   :marks  :=  67;
END;
```

Bind Variables

Displaying the value in a bind variable:

1. To show the value of your bind variable, just give the name of your bind variable as a parameter to the DBMS_OUTPUT package's PUT LINE function.

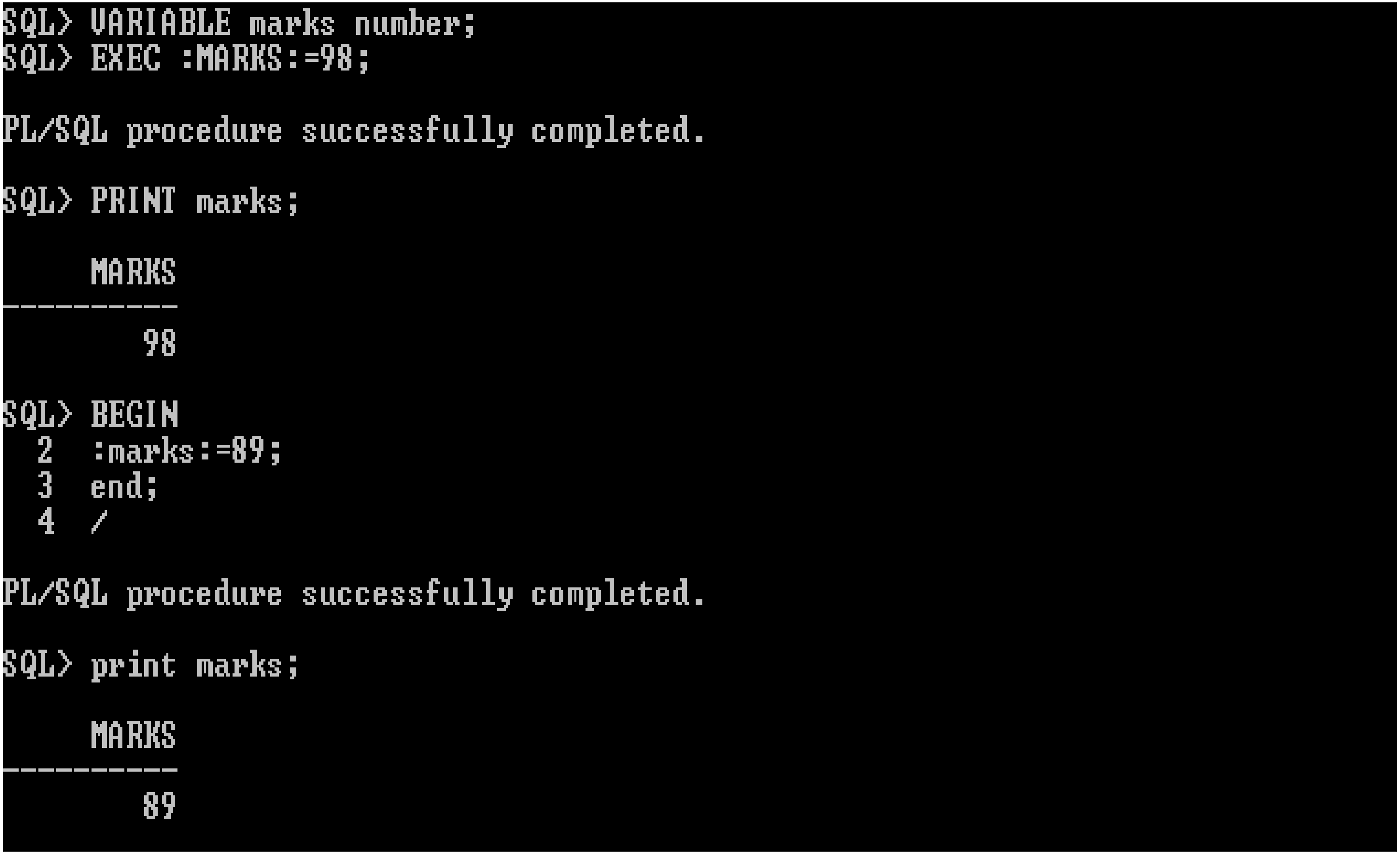
   Example: DBMS_OUTPUT.PUT LINE(**:marks**);

2. Using PRINT command,

   Syntax print variable_name;

   Example: Print marks;

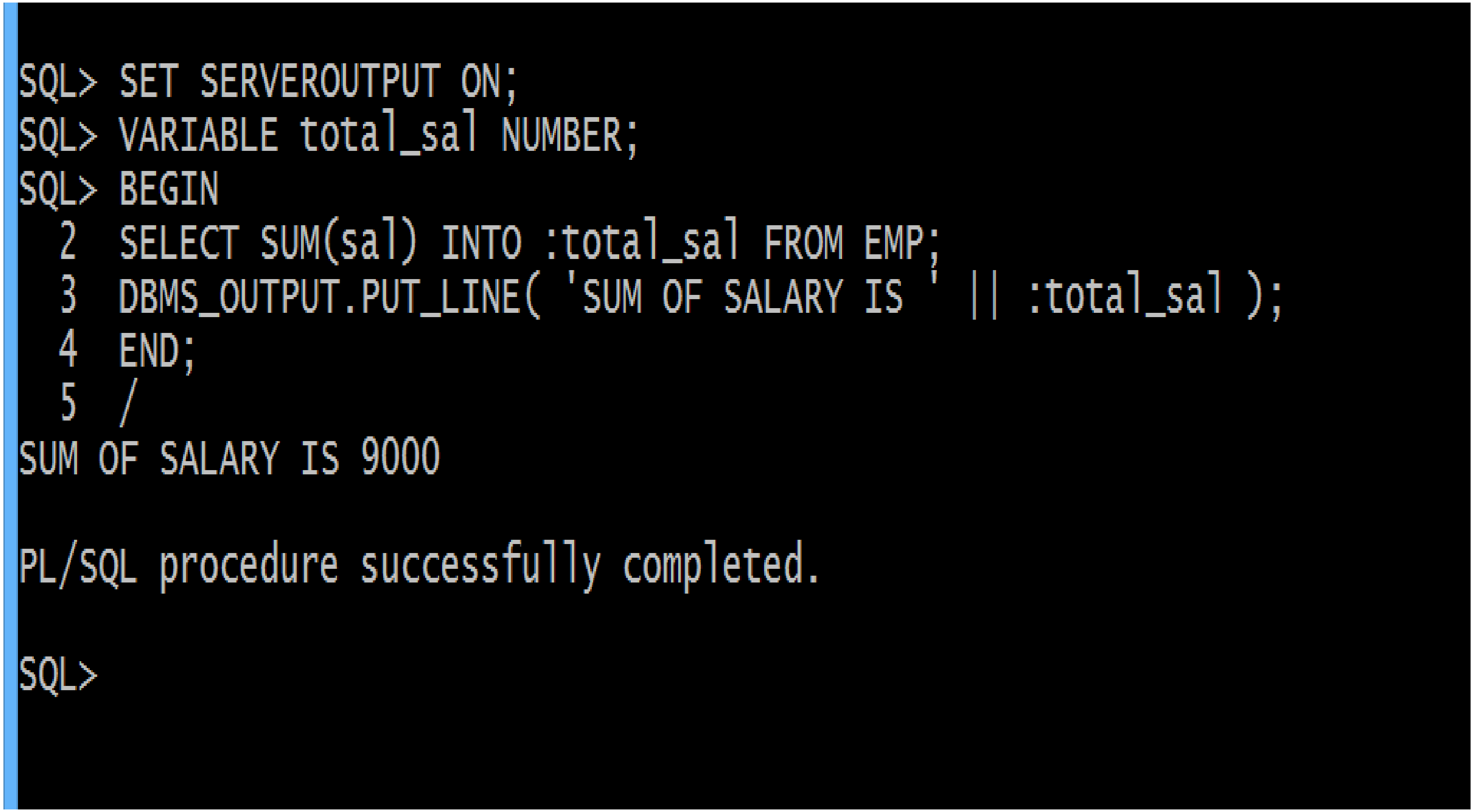
Bind Variables



```
SQL> VARIABLE marks number;
SQL> EXEC :MARKS:=98;

PL/SQL procedure successfully completed.

SQL> PRINT marks;

     MARKS
----------
        98

SQL> BEGIN
  2  :marks:=89;
  3  end;
  4  /

PL/SQL procedure successfully completed.

SQL> print marks;

     MARKS
----------
        89
```

Bind Variables

- Find total salary of all employees in PL/SQL



```
SQL> SET SERVEROUTPUT ON;
SQL> VARIABLE total_sal NUMBER;
SQL> BEGIN
     SELECT SUM(sal) INTO :total_sal FROM EMP;
     DBMS_OUTPUT.PUT_LINE( 'SUM OF SALARY IS ' || :total_sal );
     END;
     /
SUM OF SALARY IS 9000

PL/SQL procedure successfully completed.

SQL>
```

Substitution Variables

- PL/SQL blocks can accept user input while execution.
- Whenever PL/SQL processor encounters a substitution variable in a block, it waits for user input.
- All the substitution variables in the block are replaced by the inputted values before sending it to the database.
- Substitution variables are used by prefixing a variable name with the ampersand(&) or double ampersand (&&).

Substitution Variables

- **Ampersand(&)** is used when you want to give a new value each time a substitution variable is encountered in the code.
- The user has to input value even if the same substitution variable is used at multiple places in a PL/SQL program.

Substitution Variables

- **Double ampersand (&&)** is used when you want the same value to be used for all occurrences of a substitution variable in PL/SQL program.
- With **&&**, PL/SQL processor asks only once for each substitution variable. Wherever the substitution variable is encountered in the code it is replaced with previously assigned value.

- BEGIN
- DBMS_OUTPUT.PUT_LINE(‘I am going to town on ‘||’&vday’);
- DBMS_OUTPUT.PUT_LINE(‘I am coming back on ‘||‘&vday’);
- END;

- BEGIN DBMS_OUTPUT.PUT_LINE(‘I am going to town on ‘||’&&vday’); DBMS_OUTPUT.PUT_LINE(‘I am coming back on ‘||‘&vday’); END;