**Quiz**

The %TYPE attribute:

1. Is used to declare a variable according to a database column definition

2. Is used to declare a variable according to a collection of columns in a database table or view

3. Is used to declare a variable according to the definition of another declared variable

4. Is prefixed with the database table and column name or the name of the declared variable

**Answer: 1, 3, 4**

**Practice 2: Declaring PL/SQL Variables**

In this practice, you declare PL/SQL variables.

1)  Identify valid and invalid identifiers:

a) today – Valid

b) last\_name – Valid

c) today’s\_date – Invalid, character «’» cannot be used

d) Number\_of\_days\_in\_February\_this\_year – Invalid, very long identifier

e) Isleap$year – Valid

f) #number – Invalid, cannot start with character

g) NUMBER# – Valid

h) number1to7 – Valid

2)  Identify valid and invalid variable declaration and initialization:

a) number\_of\_copies PLS\_INTEGER; – Valid

b) PRINTER\_NAME constant VARCHAR2 (10); – Invalid, constant identifier must be initialized during declaration

c) deliver\_to VARCHAR2 (10):= Johnson; – Invalid, string literal must be quoted

d) By\_when DATE:=CURRENT\_DATE+1; – Valid

3) Examine the following anonymous block, and then select a statement from the following that is true. Изображение выглядит как стол

Автоматически созданное описание

a)  The block executes successfully and prints “fernandez.”

b)  The block produces an error because the fname variable is used without initializing.

c)  The block executes successfully and prints “null fernandez.”

d)  The block produces an error because you cannot use the DEFAULT keyword to initialize a variable of type VARCHAR2.

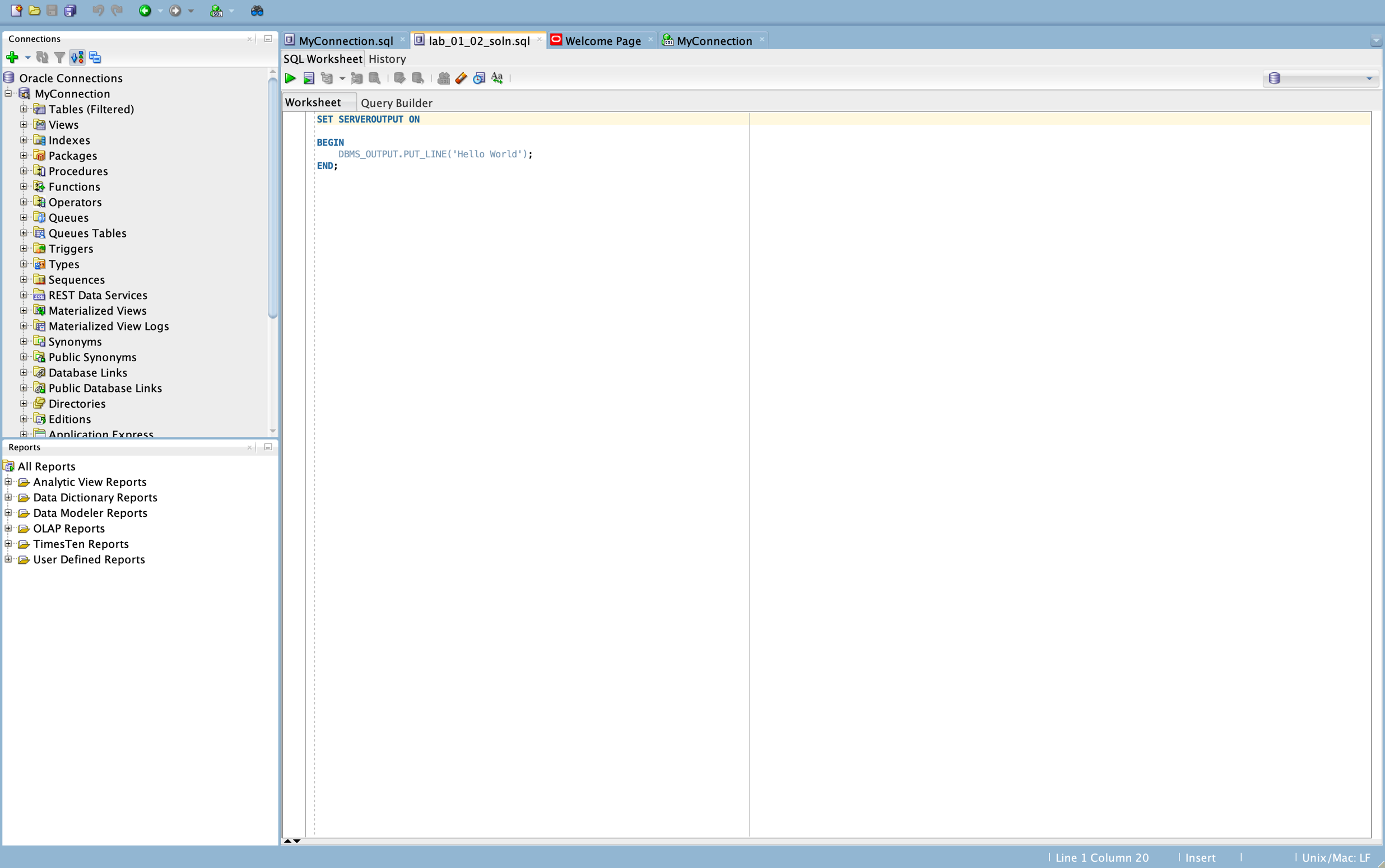
e)  The block produces an error because the v\_fname variable is not declared.

**Answer: A**

**Practice 2: Declaring PL/SQL Variables (continued)**

4)  Modify an existing anonymous block and save it as a new script.

a)  Open the lab\_01\_02\_soln.sql script, which you created in Practice 1.



**Picture 1. Open lab\_01\_02\_soln.ql**

b)  In this PL/SQL block, declare the following variables:

1. v\_today of type DATE. Initialize today with SYSDATE.

2. v\_tomorrow of type today. Use the %TYPE attribute to declare this

variable.

**SET SERVEROUTPUT ON  
  
DECLARE   
 v\_today DATE := SYSDATE;  
 v\_tomorrow v\_today%TYPE;  
BEGIN  
 DBMS\_OUTPUT.PUT\_LINE('Hello World');  
END;**

Изображение выглядит как текст

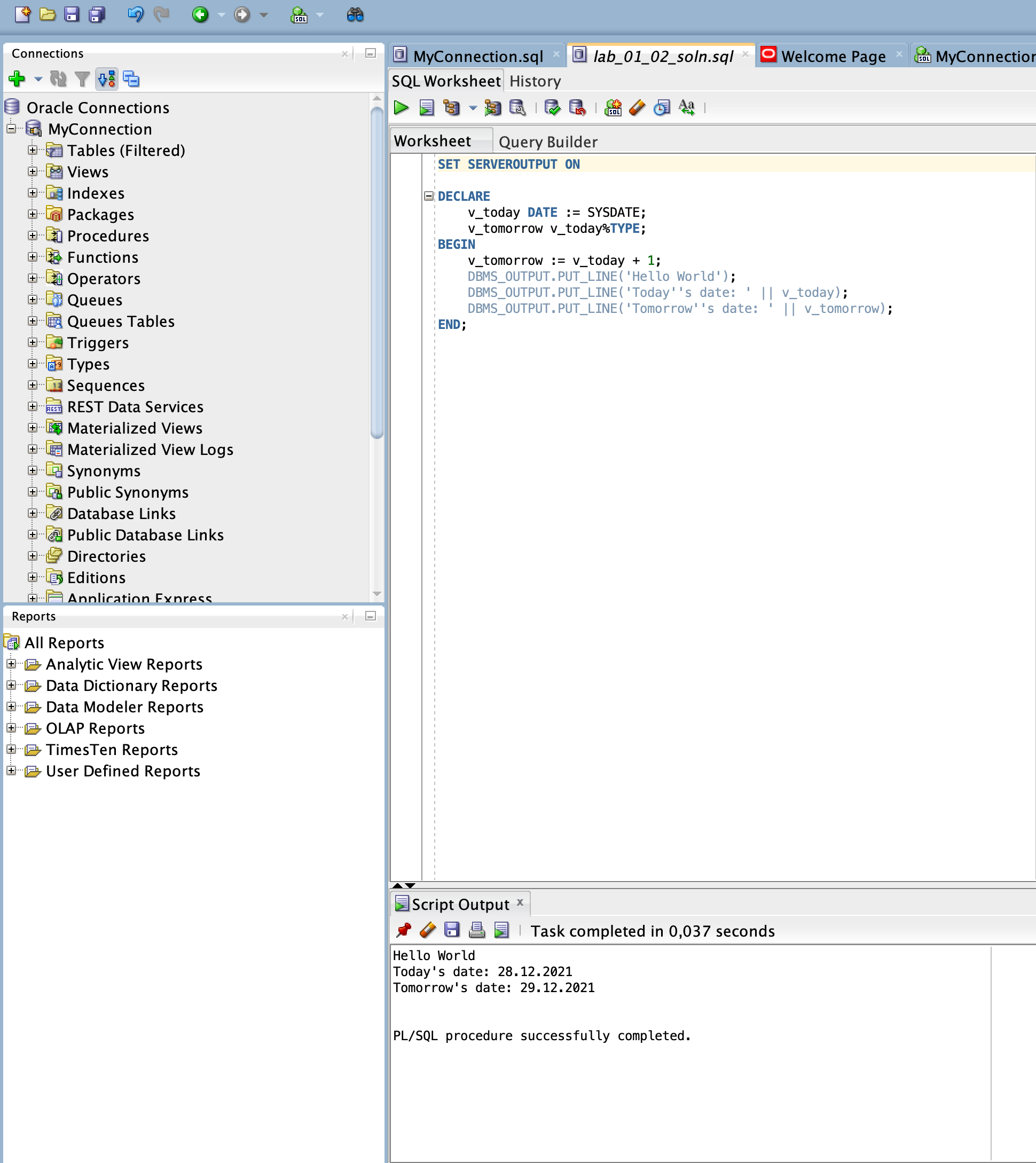
Автоматически созданное описание

c) In the executable section:

1. Initialize the v\_tomorrow variable with an expression, which calculates tomorrow’s date (add one to the value in today)

2. Print the value of v\_today and tomorrow after printing “Hello World”

**SET SERVEROUTPUT ON  
  
DECLARE  
 v\_today DATE := SYSDATE;  
 v\_tomorrow v\_today%TYPE;  
BEGIN  
 v\_tomorrow := v\_today + 1;  
 DBMS\_OUTPUT.PUT\_LINE('Hello World');  
 DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);  
 DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);  
END;**

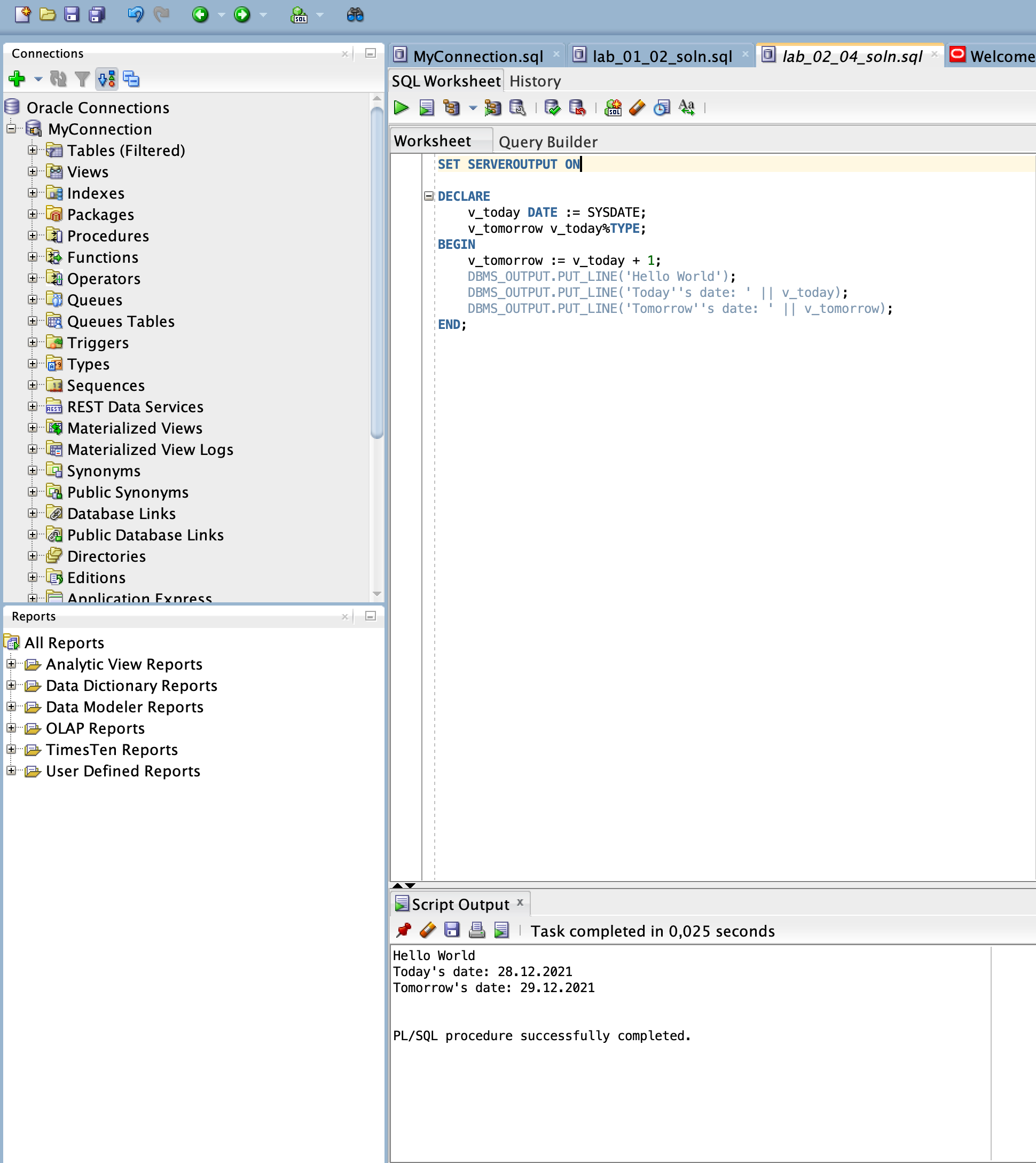


d)  Save your script as lab\_02\_04\_soln.sql, and then execute.

The sample output is as follows (the values of v\_today and v\_tomorrow will be different to reflect your current today’s and tomorrow’s date):

Изображение выглядит как стол

Автоматически созданное описание



5) Edit the lab\_02\_04\_soln.sql script.

a) Add code to create two bind variables, named b\_basic\_percent and b\_pf\_percent. Both bind variables are of type NUMBER.

**SET SERVEROUTPUT ON  
  
VARIABLE b\_basic\_percent NUMBER;  
VARIABLE b\_pf\_percent NUMBER;  
  
DECLARE  
 v\_today DATE := SYSDATE;  
 v\_tomorrow v\_today%TYPE;  
BEGIN  
 v\_tomorrow := v\_today + 1;  
 DBMS\_OUTPUT.PUT\_LINE('Hello World');  
 DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);  
 DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);  
END;**

Изображение выглядит как текст

Автоматически созданное описание

b)  In the executable section of the PL/SQL block, assign the values 45 and 12 to b\_basic\_percent and b\_pf\_percent, respectively.

**SET SERVEROUTPUT ON  
  
VARIABLE b\_basic\_percent NUMBER;  
VARIABLE b\_pf\_percent NUMBER;  
  
DECLARE   
 v\_today DATE := SYSDATE;  
 v\_tomorrow v\_today%TYPE;  
BEGIN  
 v\_tomorrow := v\_today + 1;  
   
 :b\_basic\_percent := 45;  
 :b\_pf\_percent := 12;  
   
 DBMS\_OUTPUT.PUT\_LINE('Hello World');  
 DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);  
 DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);  
END;**

Изображение выглядит как текст

Автоматически созданное описание

c)  Terminate the PL/SQL block with “/” and display the value of the bind variables by using the PRINT command.

**SET SERVEROUTPUT ON  
  
VARIABLE b\_basic\_percent NUMBER;  
VARIABLE b\_pf\_percent NUMBER;  
  
DECLARE   
 v\_today DATE := SYSDATE;  
 v\_tomorrow v\_today%TYPE;  
BEGIN  
 v\_tomorrow := v\_today + 1;  
   
 :b\_basic\_percent := 45;  
 :b\_pf\_percent := 12;  
   
 DBMS\_OUTPUT.PUT\_LINE('Hello World');  
 DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);  
 DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);  
END;  
/  
PRINT b\_basic\_percent;  
PRINT b\_pf\_percent;**

Изображение выглядит как текст

Автоматически созданное описание

d)  Execute and save your script as lab\_02\_05\_soln.sql. The sample output is as follows:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание