# Database Programming

6.Functions

## 6.Functions

A function that is stored in the database is much like a procedure, in that it is a named PL/SQL block that can take parameters and be invoked.

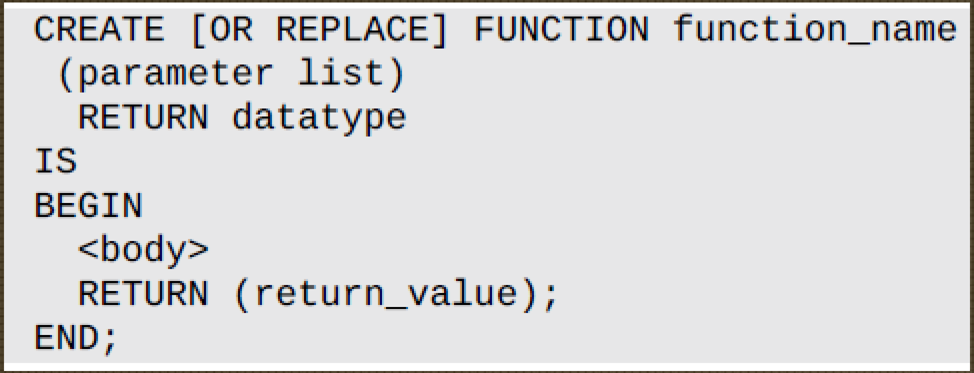
### Creating Functions

Functions are another type of stored code and are very similar to procedures. The significant difference between the two is that a function is a PL/SQL block that returns a single value. Functions can accept one, many, or no parameters, but they must have a return clause in their execution section The data type of the return value must be declared in the header of the function. A function is not a stand-alone executable in the same way that a procedure is; that is, a function must always be used in some context

## 6.Functions

### Creating Stored Functions

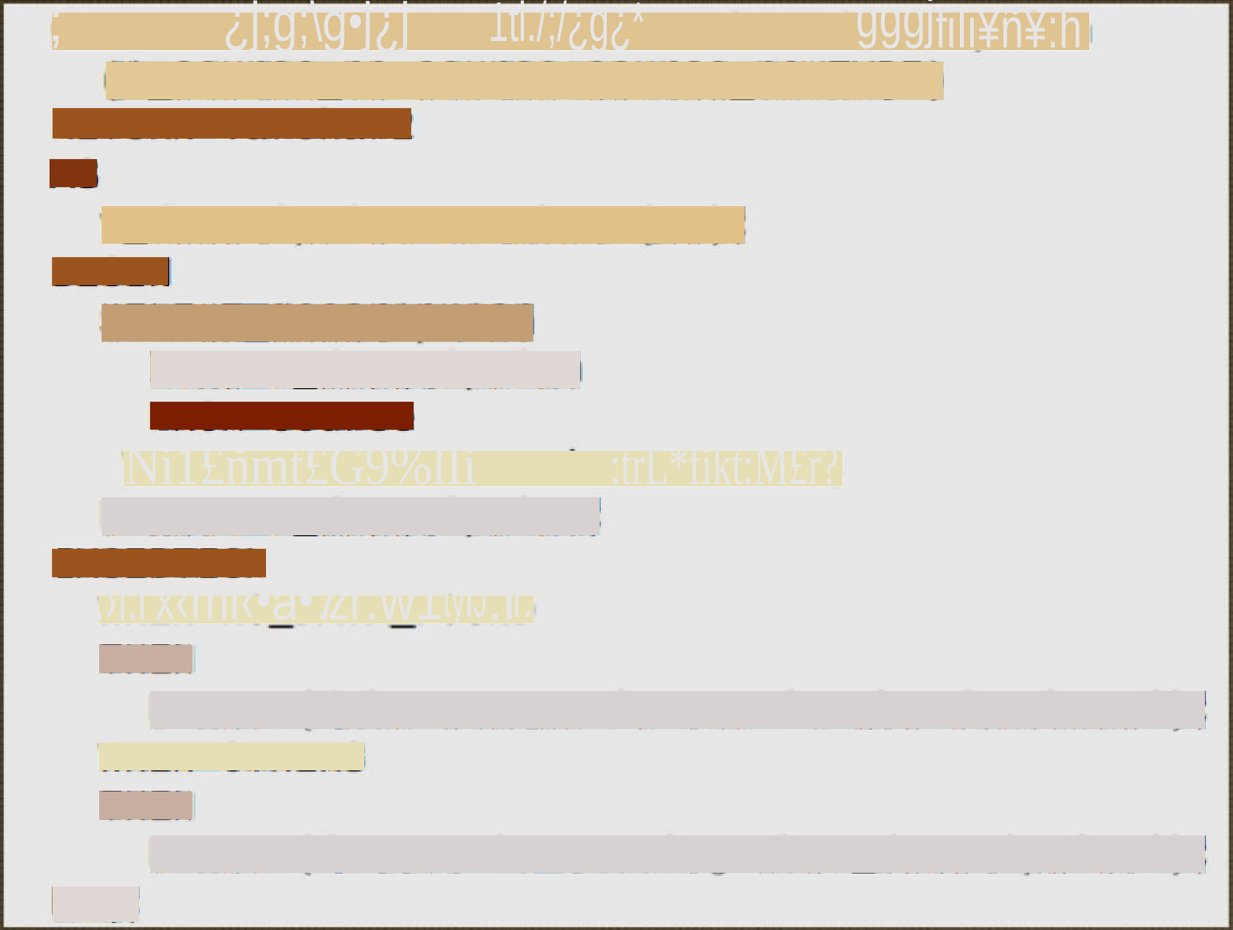
The syntax for creating a function is as follows:



```
CREATE [OR REPLACE] FUNCTION function_name
 (parameter list)
   RETURN datatype
IS
BEGIN
   <body>
   RETURN (return_value);
END;
```

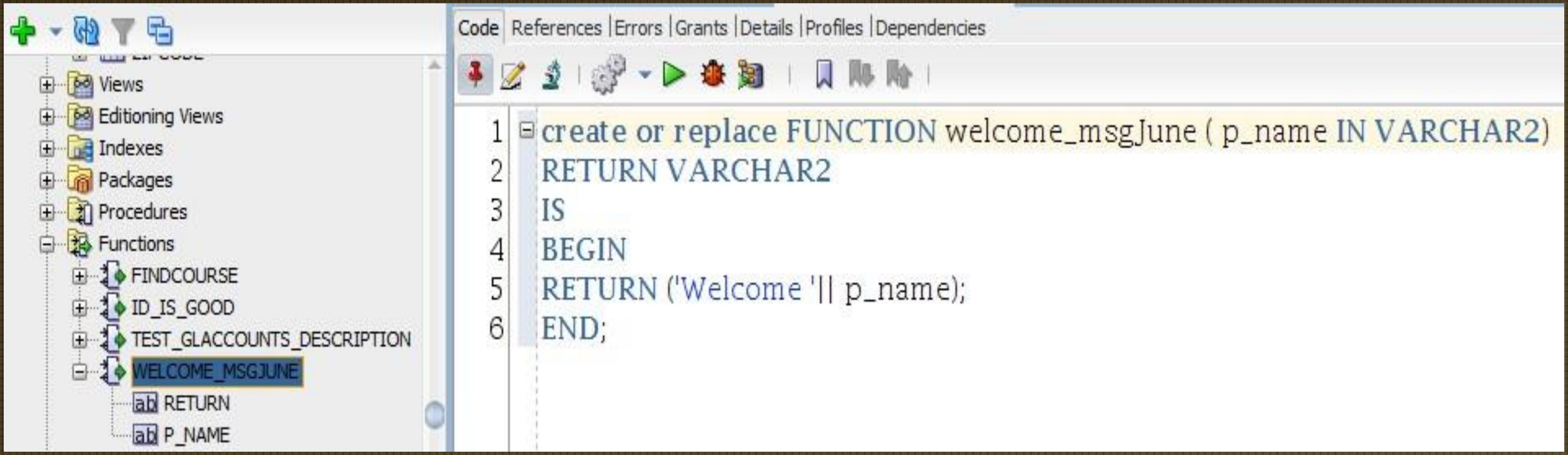
The function does not necessarily have any parameters, but it must have a RETURN value declared in the header, and it must return values for all of the possible execution streams

## 6.Functions


[illegible]

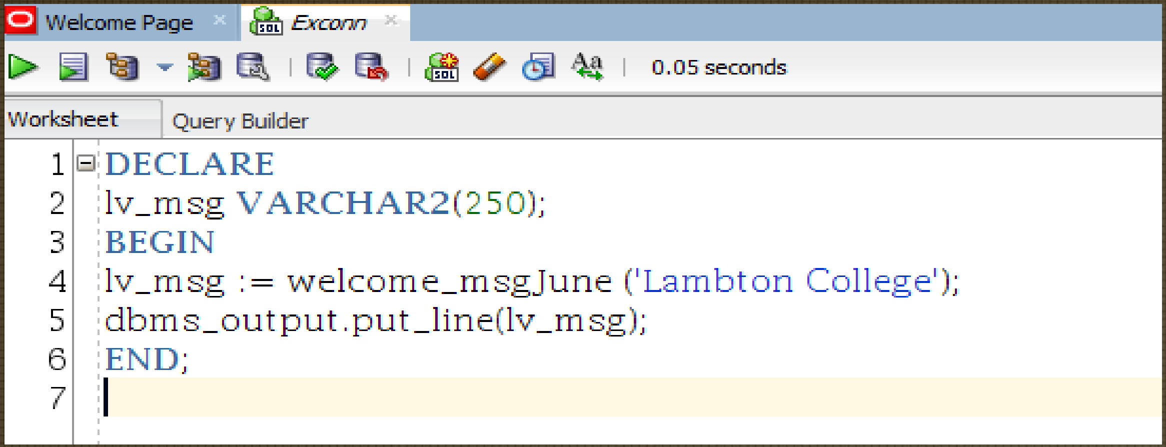
# 6.Functions

Lets consider a simple example



```
create or replace FUNCTION welcome_msgJune ( p_name IN VARCHAR2)
RETURN VARCHAR2
IS
BEGIN
RETURN ('Welcome '|| p_name);
END;
```

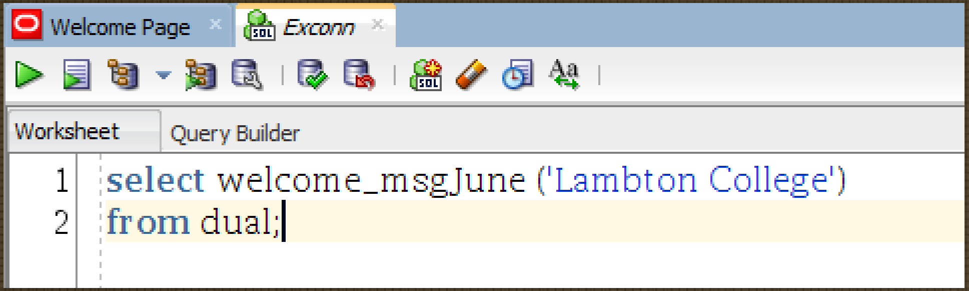
Now execute the code using following method (Method 1)



```
DECLARE
lv_msg VARCHAR2(250);
BEGIN
lv_msg := welcome_msgJune ('Lambton College');
dbms_output.put_line(lv_msg);
END;
```

# 6.Functions

Now execute the code using second method (Method 2)

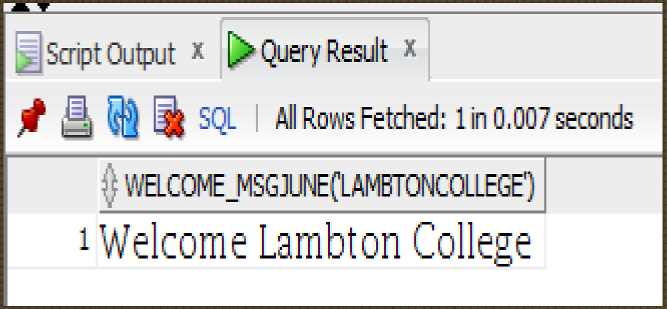


```
select welcome_msgJune ('Lambton College')
from dual;
```

Following is the output :

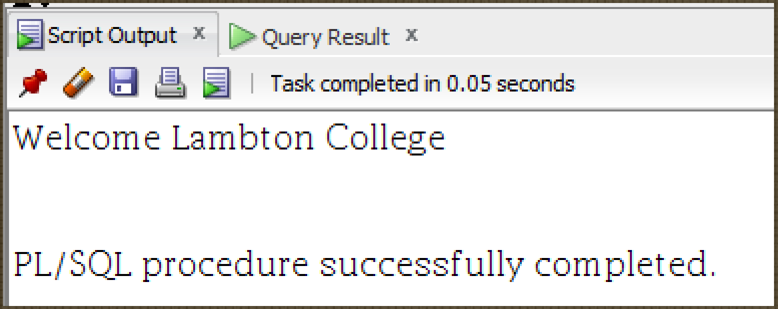
Second Method :

First Method



```
All Rows Fetched: 1 in 0.007 seconds

  WELCOME_MSGJUNE('LAMBTONCOLLEGE')
1 Welcome Lambton College
```



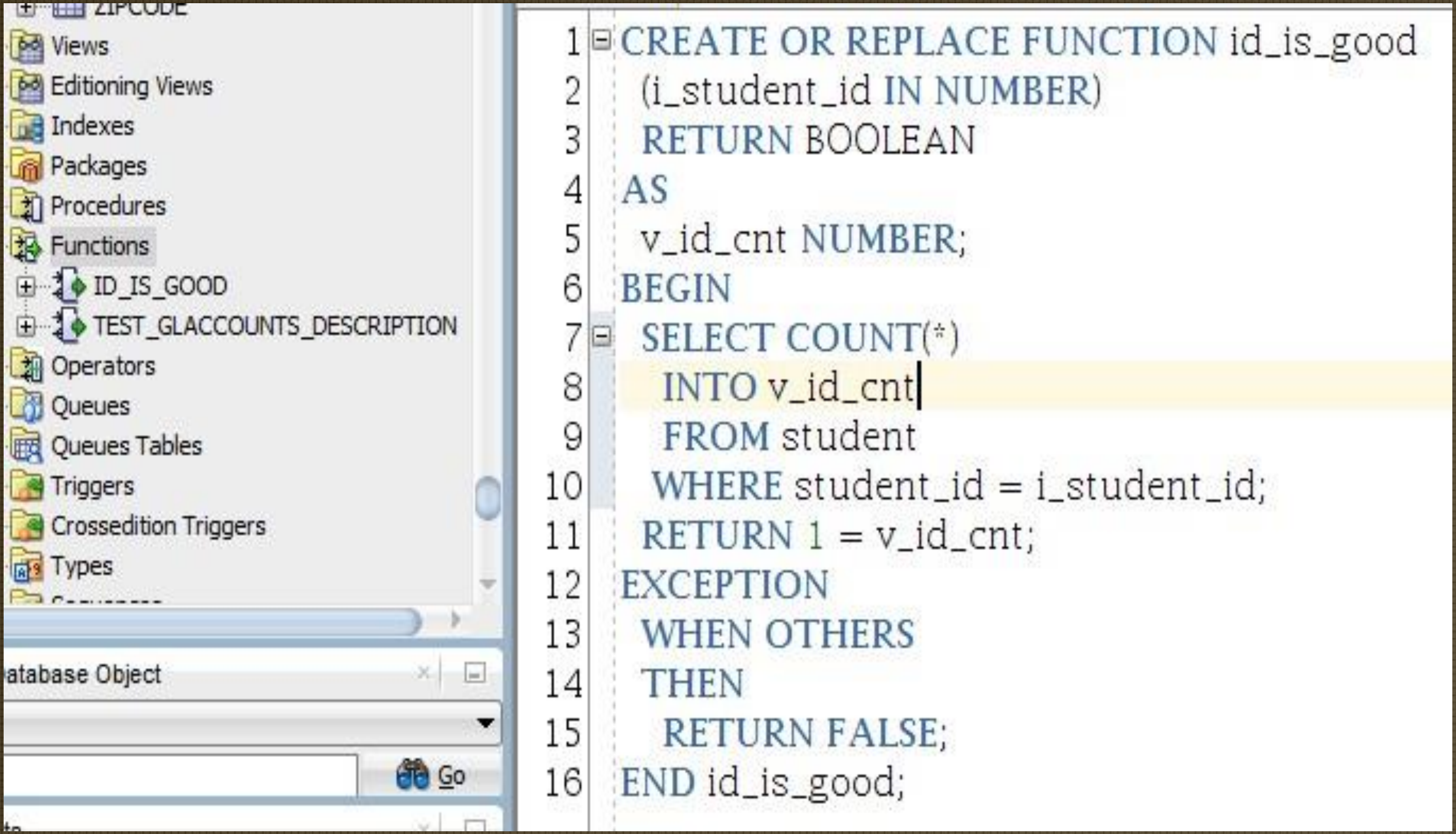
```
Task completed in 0.05 seconds

Welcome Lambton College


PL/SQL procedure successfully completed.
```

# 6.Functions

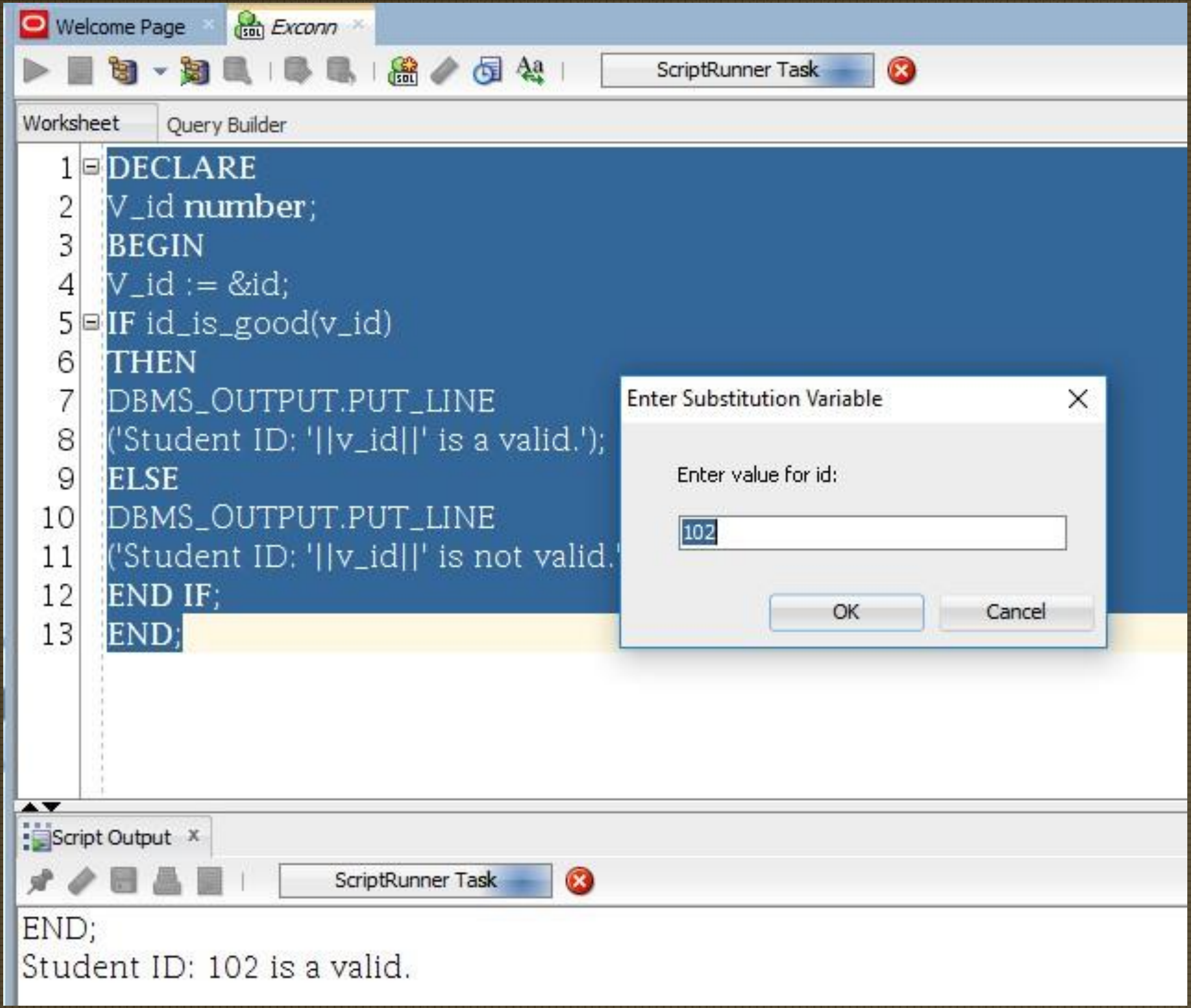
Lets consider another example makes use of the function.



```
 1 CREATE OR REPLACE FUNCTION id_is_good
 2   (i_student_id IN NUMBER)
 3   RETURN BOOLEAN
 4 AS
 5   v_id_cnt NUMBER;
 6 BEGIN
 7   SELECT COUNT(*)
 8     INTO v_id_cnt
 9     FROM student
10    WHERE student_id = i_student_id;
11   RETURN 1 = v_id_cnt;
12 EXCEPTION
13   WHEN OTHERS
14   THEN
15     RETURN FALSE;
16 END id_is_good;
```

Explain line number 11 in class

# 6.Functions

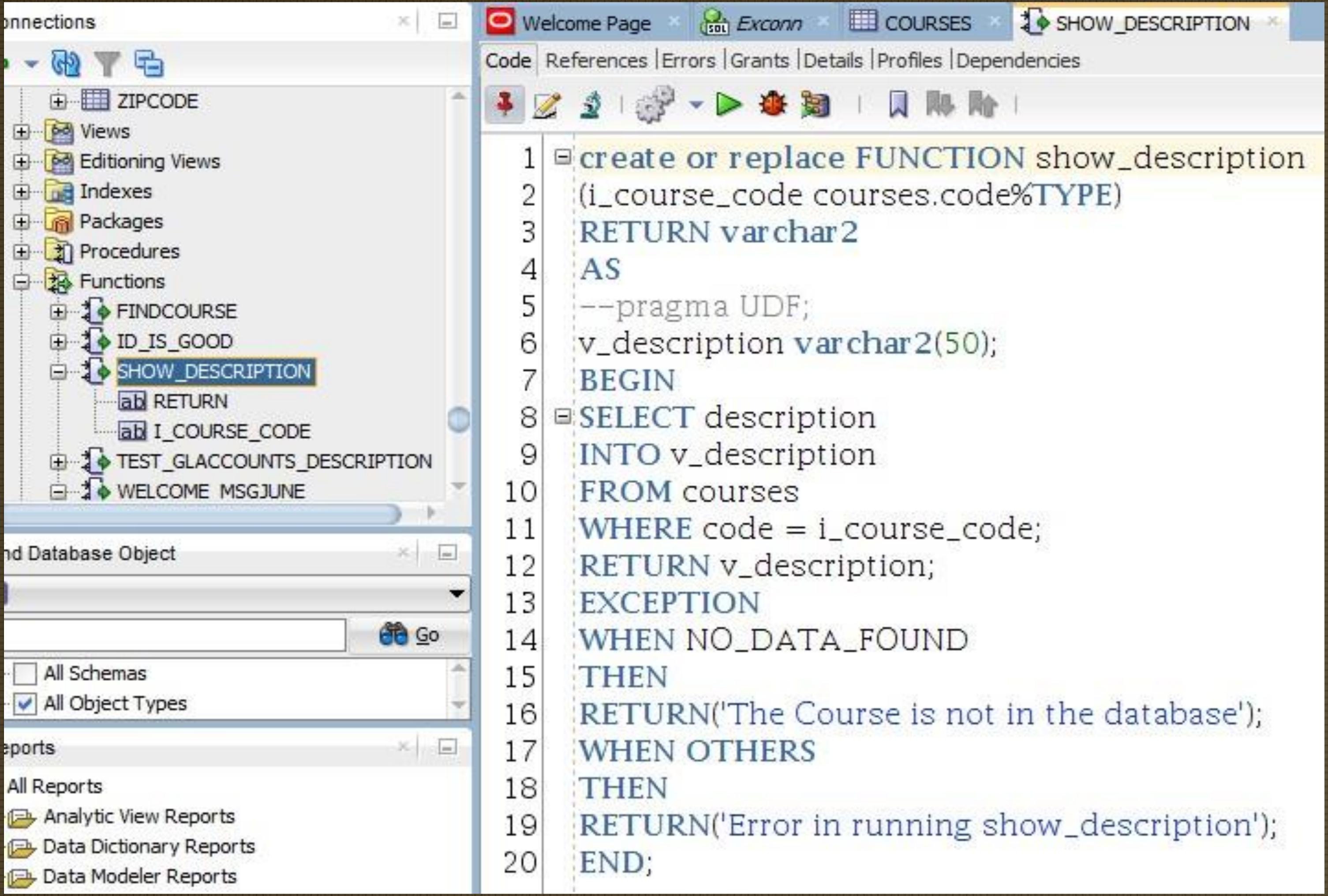


```
DECLARE
V_id number;
BEGIN
V_id := &id;
IF id_is_good(v_id)
THEN
DBMS_OUTPUT.PUT_LINE
('Student ID: '||v_id||' is a valid.');
ELSE
DBMS_OUTPUT.PUT_LINE
('Student ID: '||v_id||' is not valid.'
END IF;
END;
```

```
END;
Student ID: 102 is a valid.
```

## 6.Functions

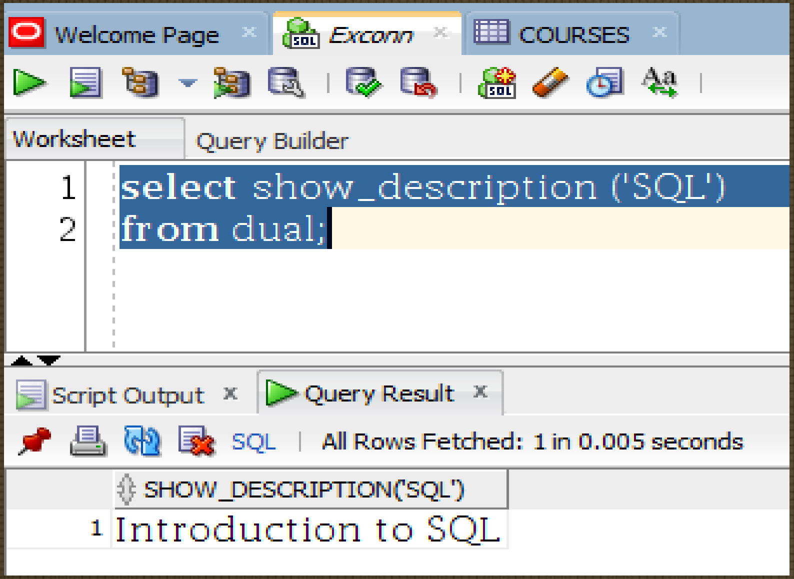
Lets consider one more example of functions.



```
create or replace FUNCTION show_description
(i_course_code courses.code%TYPE)
RETURN varchar2
AS
--pragma UDF;
v_description varchar2(50);
BEGIN
SELECT description
INTO v_description
FROM courses
WHERE code = i_course_code;
RETURN v_description;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
RETURN('The Course is not in the database');
WHEN OTHERS
THEN
RETURN('Error in running show_description');
END;
```

# 6.Functions

Following is the output :



```
select show_description ('SQL')
from dual;
```

All Rows Fetched: 1 in 0.005 seconds

```
   SHOW_DESCRIPTION('SQL')
1  Introduction to SQL
```

Q&A