# LO9.3 Create Functions

Functions are similar to procedures but they must always return a single value.

## How to create a function.

Syntax:

*CREATE [OR REPLACE] FUNCTION function\_name [parameters]*

*RETURN return\_datatype -- where the return type of the function is defined*

*IS*

*Declaration\_section*

*BEGIN*

*Execution\_section*

*Return return\_variable; --returning a value*

*EXCEPTION*

*exception section*

*Return return\_variable; --returning a value*

*END;*

Tips/Rules:

1. The return datatype can be any of the oracle datatypes like varchar, number, etc.
2. The execution and exception section both should return a value which is of the datatype defined in the header section
3. Parameters are typed the same as procedures (IN, OUT, IN OUT)

EXAMPLE:

*CREATE OR REPLACE FUNCTION employer\_details\_func*

*RETURN VARCHAR(20);*

*IS*

*emp\_name VARCHAR(20);*

*BEGIN*

*\*do some statements where you set emp\_name to a value*

*RETURN emp\_name;*

*END;*

## How to execute a PL/SQL Function?

A function can be executed in the following ways.

1. Since a function returns a value, we can assign to a variable:

*employee\_name := employer\_details\_func;*

If ‘employee\_name’ is of datatype varchar we can store the name of the employee by assigning the return type of the function to it.

1. As a part of a SELECT statement

*SELECT employer\_details\_func FROM dual;*

1. In a PL/SQL statements like:

*dbms\_output.put\_line(employer\_details\_func);*

This line displays the value returned by the function.