Creating PL/SQL Functions:

The PL/SQL Function is very similar to PL/SQL Procedure. Except the fact that it always returns a value.

Except this, all the other things of PL/SQL procedure are true for PL/SQL function too.

Syntax to create a function:

```
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];
```

Here:

* Function_name: specifies the name of the function.
* [OR REPLACE] option allows modifying an existing function.
* The optional parameter list contains name, mode and types of the parameters.
* 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. The function must contain a return statement.
* RETURN clause specifies that data type you are going to return from the function.
* Function_body contains the executable part.
* The AS keyword is used instead of the IS keyword for creating a standalone function.

Example 1 :

```
create or replace function adder(n1 in number, n2 in number)
return number
is
n3 number;
begin
n3 :=n1+n2;
return n3;
end;


DECLARE
   n3 number(2);
BEGIN
   n3 := adder(11,22);
   dbms_output.put_line('Addition is: ' || n3);
END;
```

Example 2:

```
create or replace function totalCoursesDuration
return number
is
courseDuration number;
begin
Select sum(COURSE_HOURS) into courseDuration from courses_tbl;
return courseDuration;
end;


DECLARE
   n3 number;
BEGIN
   n3 := totalCoursesDuration();
   dbms_output.put_line('Total Duration of courses is: ' || n3);
END;
```

Deleting PL/SQL Functions:

If you want to remove your created function from the database, you should use the following syntax.

Syntax:

DROP FUNCTION function_name;

PL/SQL Functions MCQs:

1. What is TRUE about PL/SQL functionalities?

1. Conditions and loops are fundamental elements of procedural languages like PL/SQL.
2. Various types and variables can be declared, as can procedures and functions, as well as types and variables of those types.
3. **Arrays can be used with it as well as handling exceptions (runtime errors).**
4. All of the above

2. AS Keyword is used in the PL/SQL function in order to create a –

1. Identity Function
2. Quadratic Function
3. **One to One Function**
4. Standalone Function

3. Which of the following is NOT present in the syntax of PL/SQL function?

1. Function_name
2. [OR REPLACE]
3. **Optional Parameter List**
4. None of the above

4. The call function returns program control to the ____ after successful completion of the defined task.

1. Main Table
2. Main Database
3. **Main Row**
4. Main Program

5. It is called ____ when the subprogram calls itself and ____ is the process.

1. Recursion, Recursive Call
2. Recursive Call, Recursion
3. Recursive Name, Recursive Call
4. Recursive Call, Recursive Nam