**Sub Programs:** Subprograms are named PL/SQL blocks that can take parameters and be invoked. PLSQL has two types of subprograms, Procedures and functions.

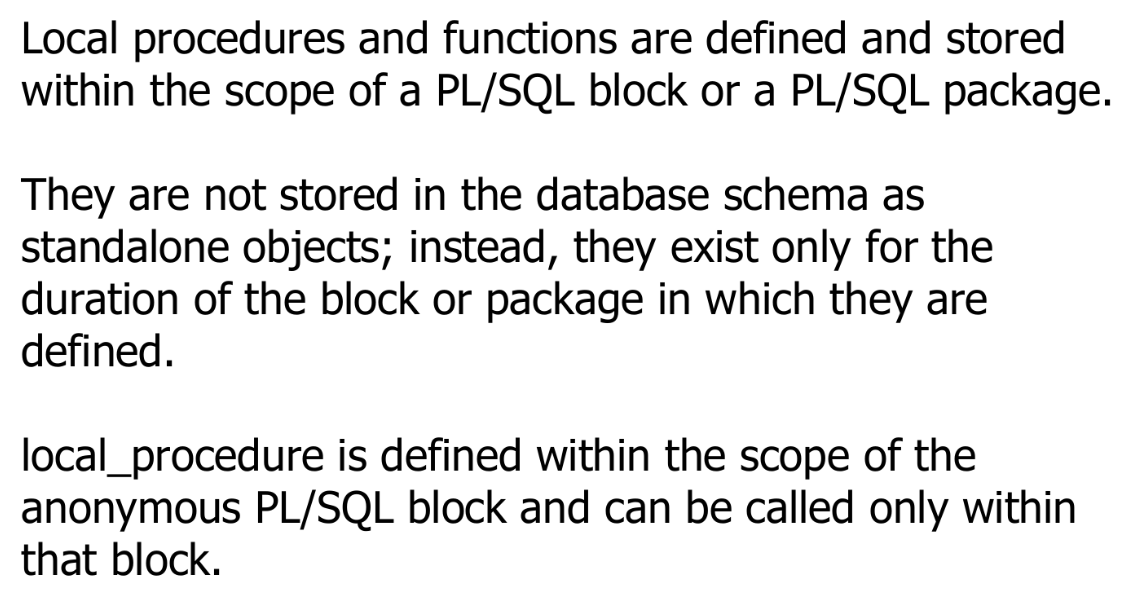
**Procedures:**

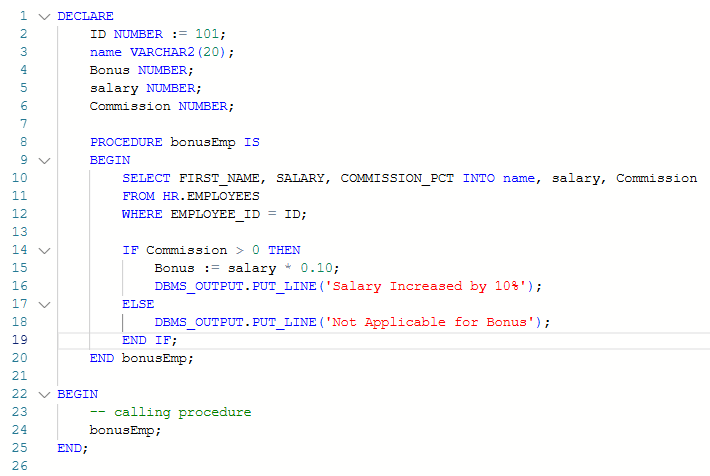
a **procedure** is a named PL/SQL block that performs one or more actions. Unlike a function, a procedure does not necessarily return a value to its caller (though it can return values through OUT parameters). Procedures are excellent for encapsulating business logic, performing DML operations (INSERT, UPDATE, DELETE), and managing transactions.

**Two Types of Procedures:**

***1. Local Procedure 2. Stored Procedure***

**1. Local Procedure:**

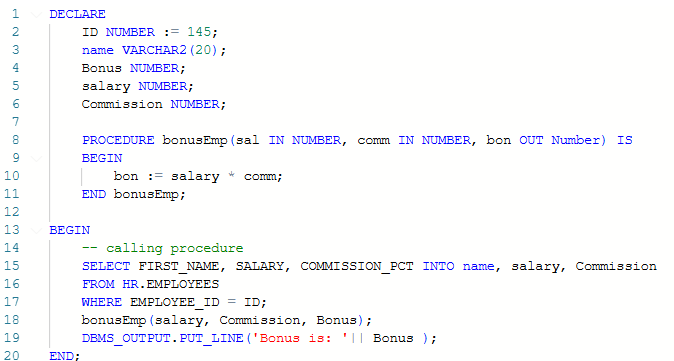




A white background with black text

AI-generated content may be incorrect.

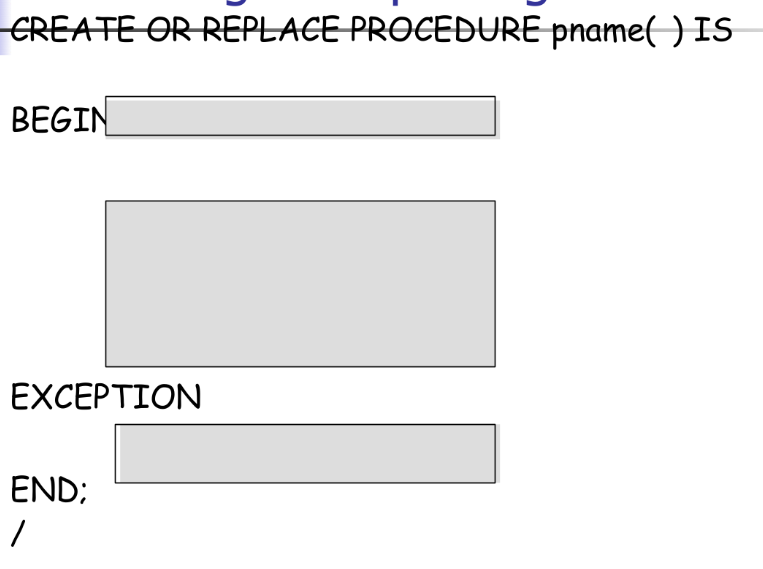
**USING IN & OUT PARAMETERS in Local Procedure**





**STORED PROCEDURE:**

A stored procedure is a precompiled collection of SQL statements and optional control-of-flow statements stored under a name and processed as a unit.



**Creating a Stored Procedure:**

A white background with black text

AI-generated content may be incorrect.

**Calling a Stored Procedure:**

A black and white text

AI-generated content may be incorrect.

**=> Operator (Association Operator):**

In PL/SQL, the **=>** operator is known as the **association operator** or **named notation** and is used for passing arguments to procedures and functions. A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Functions:**

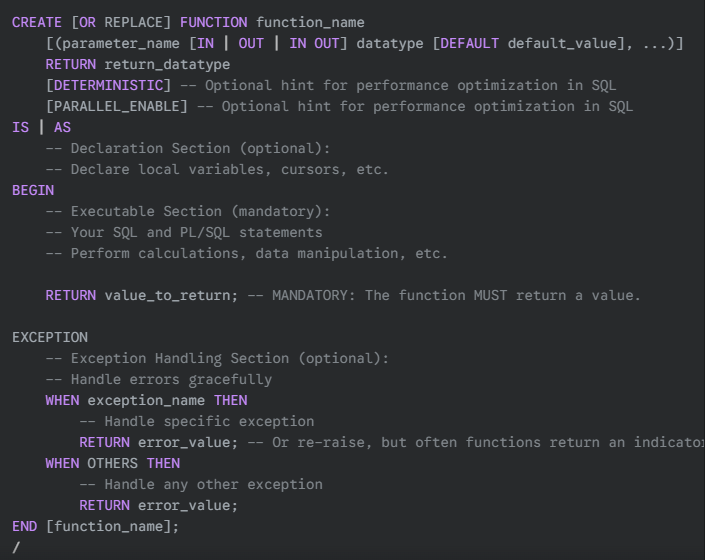
A named PL/SQL block that performs a specific set of operations.

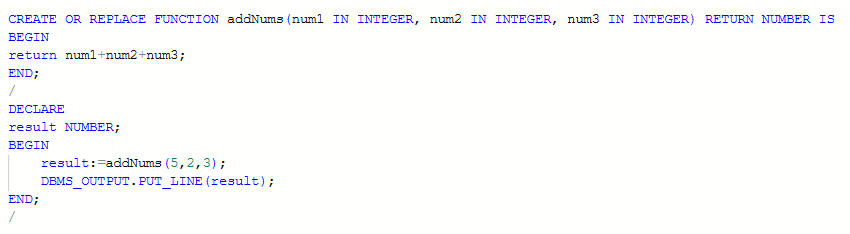
It computes and returns a single value of a specified data type.

Can accept zero or more input parameters.

Can be called as part of an expression, like in a SELECT statement, WHERE clause, or an assignment.

A close-up of a message

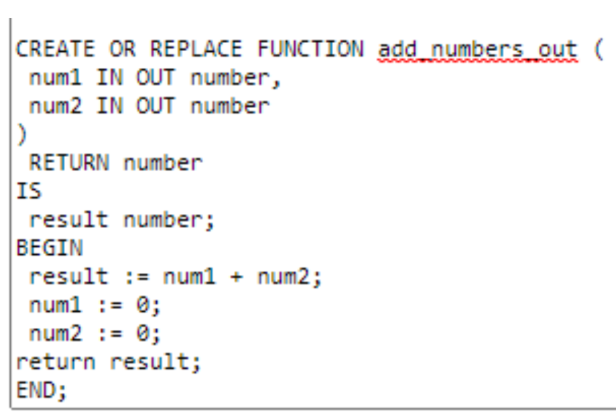
AI-generated content may be incorrect.

**Red Rectangle 🡪** Calling the function with parameter and getting the value which is returned by the function in a variable

**PASS BY VALUE PARAMETERS:**

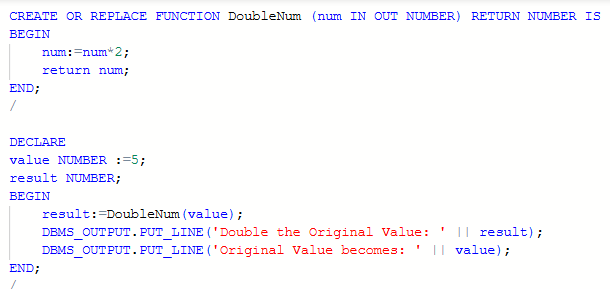
* In PL/SQL, parameters are always passed by value.
* But when a parameter is declared as IN OUT (or OUT), it essentially allows the function or procedure to modify the value of the parameter.
* A screenshot of a computer program

  AI-generated content may be incorrect.And these modifications are visible outside the function or procedure.



**PASS BY REFERENCE PARAMETERS:**

* The pass-by-reference function demonstrates returning a value while altering the input variable. Parameters values have been changed.
* A write-only (OUT mode) formal parameter can’t work in this type of call because the new value is never read.
* If you want to keep original values of the variables unchanged, you may want to reconsider whether using IN OUT parameters is appropriate for your specific use case.



A black text on a white background

AI-generated content may be incorrect.

**Packages:**

* Packages are a group of procedures, functions, variables, constants, cursors
* and SQL statements grouped together into a single unit.
* Similar types of Procedures/Functions are placed in one package.
* Supports Function Overloading
* Whole Package Can not be executed, only members can be.
* Procedures/Functions defined in a package are private to it.

**Component of an Oracle Package:**

* Package specification
* Package Body

**Package Specification (or Header):**

* This is the interface to the package.
* It declares all the public (visible and accessible from outside the package) items: procedures, functions, cursors, variables, constants, exceptions, and types.
* It tells you what the package offers, but not how it's implemented.
* It does not contain any executable code for procedures or functions, only their signatures (name, parameters, return type for functions).

A close-up of a box

AI-generated content may be incorrect.

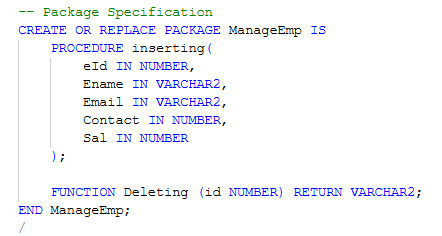
**Package Body (or Definition):**

* This is the implementation of the package.
* It contains the actual executable code for all the public subprograms declared in the specification.
* It can also contain private (internal to the package) subprograms, variables, constants, cursors, etc., that are not exposed outside the package.
* It tells you how the package implements its functionality.

A white box with red and green text

AI-generated content may be incorrect.

CODE:

 A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.