 **LENGTH**

* **Definición corta**: Devuelve la longitud de una cadena de caracteres.
* **Sintaxis**: LENGTH(<cadena>)  
  Ejemplo:

resultado := LENGTH('HOLA MUNDO'); -- Devuelve 10

 **INSTR**

* **Definición corta**: Busca la posición de una subcadena dentro de una cadena.
* **Sintaxis**: INSTR(<cadena>, <subcadena>, <posición\_inicio>, <ocurrencia>)
  + <cadena>: Cadena principal donde se realiza la búsqueda.
  + <subcadena>: Texto a buscar.
  + <posición\_inicio> (opcional): Posición desde la cual se comienza a buscar.
  + <ocurrencia> (opcional): Número de la ocurrencia específica a buscar.  
    Ejemplo:

sql

CopiarEditar

resultado := INSTR('AQUI ES DONDE SE BUSCA', 'BUSCA', 1, 1); -- Devuelve 18

**REPLACE**

* **Definición corta**: Reemplaza un texto específico dentro de una cadena por otro texto.
* **Sintaxis**:

REPLACE(<expresión>, <búsqueda>, <reemplazo>)

* + <expresión>: La cadena original donde se hará el reemplazo.
  + <búsqueda>: El texto que deseas encontrar y reemplazar.
  + <reemplazo>: El texto que sustituirá al encontrado.
* **Ejemplo**:

resultado := REPLACE('HOLA MUNDO', 'HOLA', 'VAYA'); -- Devuelve 'VAYA MUNDO'

SUBSTR(<expresion>, <posicion\_ini>, <longitud> )

* **Definición corta**: Extrae una parte de una cadena, empezando desde una posición específica y con una longitud determinada.
* **Sintaxis**:
  + <expresión>: La cadena de la que se extraerá una porción.
  + <posición\_inicio>: La posición inicial desde donde comienza la extracción (comienza en 1).
  + <longitud>: (Opcional) Número de caracteres a extraer. Si no se indica, devuelve desde la posición de inicio hasta el final de la cadena.
* **Ejemplo**:

resultado := SUBSTR('HOLA MUNDO', 6, 5); -- Devuelve 'MUNDO'

**1. Procedimientos y Funciones en PL/SQL**

**Procedimientos**

Un procedimiento en PL/SQL es un **subprograma** que contiene un conjunto de instrucciones que se ejecutan al ser invocadas. Un **procedimiento** no devuelve ningún valor, pero puede realizar tareas como insertar datos, actualizar registros, etc.

1. Locales con créate
2. En memoria solo procedure

**Sintaxis de un procedimiento:**

CREATE [OR REPLACE] PROCEDURE nombre\_procedimiento

(param1 IN tipo\_dato, param2 OUT tipo\_dato) -- Parámetros

IS

-- Declaración de variables locales

BEGIN

-- Instrucciones de la lógica del procedimiento

-- Puede hacer uso de las variables y parámetros

END nombre\_procedimiento;

**Componentes de un procedimiento:**

* **nombre\_procedimiento**: Es el nombre con el que se invoca el procedimiento.
* **Parámetros(PARAMETROS FORMALES)**: Son como **variables** que se definen dentro del procedimiento y pueden recibir valores cuando se llama al procedimiento. Los parámetros pueden ser:
  + **IN**: Pasan un valor al procedimiento.
  + **OUT**: El procedimiento puede devolver un valor a través de este parámetro.
  + **IN OUT**: Permiten tanto recibir un valor como devolver un valor al exterior.
* **PARAMETROS REALES**: toman el valor de los parámetros formales y ejecutan

**Ejemplo de procedimiento:**

Supongamos que queremos crear un procedimiento para registrar un cliente. Este procedimiento tomará un ID, un Nombre y un Edad para registrar un cliente y devolver la edad modificada.

CREATE OR REPLACE PROCEDURE registrar\_cliente (

p\_id IN NUMBER, -- Parámetro de entrada

p\_nombre IN VARCHAR2, -- Parámetro de entrada

p\_edad OUT NUMBER -- Parámetro de salida

) IS

BEGIN

-- Aquí iría la lógica de registro de cliente

p\_edad := 30; -- Se asigna un valor a p\_edad como ejemplo

DBMS\_OUTPUT.PUT\_LINE('Cliente ' || p\_nombre || ' registrado con ID ' || p\_id);

END registrar\_cliente;

Este procedimiento, cuando se llama, registra un cliente y devuelve la edad a través del parámetro p\_edad.

**Invocar el procedimiento:**

EXECUTE registrar\_cliente(1, 'Juan', 30); -- Llamada con parámetros IN y OUT

**Funciones**

Una **función** en PL/SQL es un **subprograma** que realiza un cálculo o acción y **devuelve un valor**. A diferencia de los procedimientos, las funciones sí devuelven un valor que puede ser utilizado dentro de una expresión o asignado a una variable.

**Sintaxis de una función:**

**CREATE [OR REPLACE] FUNCTION nombre\_funcion**

**(param1 IN tipo\_dato, param2 IN OUT tipo\_dato)**

**RETURN tipo\_valor -- Indica el tipo de valor que va a devolver la función**

**IS**

**-- Declaración de variables locales**

**BEGIN**

**-- Lógica de la función**

**RETURN valor\_a\_devolver; -- El valor que devuelve la función**

**END nombre\_funcion;**

**Componentes de una función:**

* **nombre\_funcion**: Es el nombre que se utilizará para invocar la función.
* **RETURN tipo\_valor**: Define el tipo de dato que la función devolverá (por ejemplo, NUMBER, VARCHAR2).
* **Lógica**: Aquí se colocan las instrucciones para realizar los cálculos o la acción. Al final, se debe usar la palabra clave RETURN para devolver el valor.

**Ejemplo de función:**

Vamos a crear una función para calcular la edad de una persona basada en su fecha de nacimiento:

CREATE OR REPLACE FUNCTION obtener\_edad (

p\_fecha\_nacimiento IN DATE -- Parámetro de entrada

) RETURN NUMBER IS

v\_edad NUMBER; -- Variable local para almacenar la edad calculada

BEGIN

v\_edad := EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM p\_fecha\_nacimiento);

RETURN v\_edad; -- Devuelve el valor calculado de la edad

END obtener\_edad;

Esta función calcula la edad de una persona a partir de su fecha de nacimiento.

**Invocar la función:**

SELECT obtener\_edad(TO\_DATE('1990-05-25', 'YYYY-MM-DD')) FROM dual;

El valor calculado por la función será devuelto como resultado de la consulta.

**2. Funciones Predefinidas en PL/SQL**

**Funciones de Caracteres:**

PL/SQL ofrece varias funciones para trabajar con cadenas de texto.

* **LENGTH**: Devuelve la longitud de una cadena de caracteres.

SELECT LENGTH('Hola Mundo') FROM dual; -- Resultado: 10

* **INSTR**: Busca una subcadena dentro de otra y devuelve la posición.

SELECT INSTR('Hola Mundo', 'Mundo') FROM dual; -- Resultado: 6

* **REPLACE**: Reemplaza una subcadena por otra.

SELECT REPLACE('Hola Mundo', 'Mundo', 'PL/SQL') FROM dual; -- Resultado: Hola PL/SQL

**Funciones Numéricas:**

* **MOD**: Devuelve el resto de la división entre dos números.

SELECT MOD(10, 3) FROM dual; -- Resultado: 1

* **ROUND**: Redondea un número al entero más cercano.

SELECT ROUND(15.67) FROM dual; -- Resultado: 16

**Funciones de Fechas:**

* **SYSDATE**: Devuelve la fecha y hora actuales del sistema.

SELECT SYSDATE FROM dual;

* **TRUNC**: Elimina la parte de la fecha que no es relevante (como la hora).

SELECT TRUNC(SYSDATE) FROM dual; -- Resultado: Fecha sin hora

**Funciones de Conversión:**

* **TO\_DATE**: Convierte una cadena en una fecha, usando un formato específico.

SELECT TO\_DATE('2025-01-01', 'YYYY-MM-DD') FROM dual;

* **TO\_CHAR**: Convierte un valor a una cadena de texto.

SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD') FROM dual;

**3. Procedimientos y Funciones Almacenados**

Cuando se **almacenan procedimientos o funciones** en la base de datos, pueden ser reutilizados muchas veces sin tener que redefinirlos cada vez.

**Creación de Procedimientos Almacenados:**

CREATE OR REPLACE PROCEDURE procedimiento\_ejemplo

IS

BEGIN

-- Lógica del procedimiento

END procedimiento\_ejemplo;

**Invocar Procedimientos Almacenados:**

Se pueden invocar directamente desde SQL\*Plus:

EXECUTE procedimiento\_ejemplo;

**Creación de Funciones Almacenadas:**

CREATE OR REPLACE FUNCTION funcion\_ejemplo

RETURN NUMBER

IS

BEGIN

RETURN 42;

END funcion\_ejemplo;

**Invocar Funciones Almacenadas:**

Las funciones almacenadas pueden invocarse en consultas SQL:

SELECT funcion\_ejemplo FROM dual;

**4. Notación para Pasar Parámetros**

Cuando invocas un procedimiento o función, puedes pasar los parámetros de diferentes maneras:

* **Posicional**: Se pasan en el orden en que están definidos.

EXECUTE procedimiento\_ejemplo(10, 'Juan');

* **Nominal**: Se pasan por **nombre y valor**.

EXECUTE procedimiento\_ejemplo(p\_id => 10, p\_nombre => 'Juan');

Espero que esta explicación más detallada te ayude a comprender mejor los **procedimientos** y **funciones** en PL/SQL, cómo se crean, cómo se invocan y cómo usar las funciones predefinidas. Si tienes dudas o necesitas ejemplos adicionales, no dudes en preguntar.