CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Programación de Bases de Datos II

funciones y procedimientos almacenados

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Rutinas

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Rutinas

- *Bloques de código PL/pgSQL, referenciados bajo un nombre, que realizan una acción determinada.*
- *Aceptan parámetros.*
- *Pueden ser invocados.*
- *Pueden estar almacenados en la base de datos.*
- *Dos tipos:*
    - *Funciones*
    - *Procedimientos*

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Funciones

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Predefinidas en PostgreSQL

PostgreSQL provee un gran número de funciones predefinidas, que ya hemos utilizado en ocasiones anteriores

```
SELECT
  phone_number,
  SUBSTR(phone_number, STRPOS(phone_number,'.')+1) as "NO PREFIX"
FROM employees
WHERE JOB_ID LIKE '%MAN' AND LENGTH(phone_number)>12
ORDER BY phone_number;
```

¡¡ Siempre devuelven un valor !!

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Funciones de usuario
Sintaxis

La sentencia `CREATE FUNCTION` le permite definir una nueva función definida por el usuario.

```
create [or replace] function nombre_función(lista_parámetros)
returns tipo_retorno
language plpgsql
as
$$
declare
    -- declaración de variables
begin
    -- lógica de la función
end;
$$
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Ejemplo de funciones

Cuenta los agentes cuya categoría se encuentra entre los parámetros `cat_desde` y `cat_hasta`.

```
create function get_agentes_count(cat_desde int, cat_hasta int)
returns int
language plpgsql
as
```

ENCABEZADO

```
$$
declare
    contador_agentes integer;
begin
    select count(*)
    into contador_agentes
    from agentes
    where categoria between cat_desde and cat_hasta;

    return contador_agentes;
end;
$$;
```

CUERPO

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Invocar funciones

Usaremos la notación posicional.

```
select get_agentes_count(1, 2);

 get_agentes_count
-------------------
                 9
(1 fila)

0

select get_agentes_count(2, 3);

 get_agentes_count
-------------------
                 3
(1 fila)
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Modos de parámetros

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Modos de parámetros

IN	OUT	INOUT
El valor por defecto	Especificado explícitamente	Especificado explícitamente
Pasar un valor a la función	Devolver un valor de una función	Pase un valor a una función y devuelva un valor actualizado.
los parámetros `in` actúan como constantes	los parámetros actúan `out` como variables no inicializadas	los parámetros `inout` actúan como variables inicializadas
No se le puede asignar un valor	Debe asignar un valor	Se le debe asignar un valor

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Ejemplo IN

Encuentra una oficina por su **identificador** y devuelve su **domicilio**

```
create or replace function find_oficina_by_id(p_oficina_id int)
returns varchar
language plpgsql
as $$
declare
    oficina_domicilio oficinas.domicilio%type;
begin
    -- encuentra el domicilio de la oficina por su identificador
    select domicilio
    into oficina_domicilio
    from oficinas
    where identificador = p_oficina_id;

    if not found then
        raise 'Oficina con el identificador % no encontrada', p_oficina_id;
    end if;
    return oficina_domicilio;
end;$$;
```

```
select find_oficina_by_id(1);
 find_oficina_by_id
--------------------
 Gran vía, 37
(1 fila)
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Ejemplo OUT

Encuentra una oficina por su **identificador** y devuelve su **domicilio**

```
create or replace function get_datos_oficina(
    in p_oficina_id int,
    out p_nombre varchar(40),
    out p_domicilio varchar(40),
    out p_codigo_postal varchar(5)
)
language plpgsql
as $$
begin
    -- encuentra los datos de la oficina por su identificador
    select nombre, domicilio, codigo_postal
    into p_nombre, p_domicilio, p_codigo_postal
    from oficinas
    where identificador = p_oficina_id;
    if not found then
        raise 'Oficina con el identificador % no encontrada', p_oficina_id;
    end if;
end;$$;
```

```
select * from get_datos_oficina(1);

 p_nombre | p_domicilio  | p_codigo_postal
----------+--------------+-----------------
 Madrid   | Gran vía, 37 | 28000
(1 fila)
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Procedimientos

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Diferencia con funciones

- Los procedimientos se definen con el comando CREATE PROCEDURE, no CREATE FUNCTION.
- No devuelven ningún valor (~~RETURNS~~), aunque pueden utilizar parámetros output.
- Se invocan usando el comando CALL
- Puede utilizar transacciones.

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Sintaxis

La sentencia `CREATE PROCEDURE` le permite definir un nuevo procedimiento.

```
create [or replace] procedure nombre_procedimiento(lista_parámetros)
language plpgsql
as $$
declare
    -- declaración de variables
begin
    -- cuerpo del procedimiento almacenado
end; $$
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Ejemplo

```
create or replace procedure inserta_agente(
    p_identificador int,
    p_nombre varchar(60),
    p_usuario varchar(20),
    p_clave varchar(20),
    p_habilidad int,
    p_categoria int
)
language plpgsql
as $$
begin
    insert into agentes (identificador,nombre, usuario, clave, habilidad, categoria)
    values (p_identificador, p_nombre, p_usuario, p_clave, p_habilidad,
p_categoria);

end;$$;
```

```
call inserta_agente(2222, 'Federico Fernández', 'fedfer', 'ferfed2', 1, 1);
```