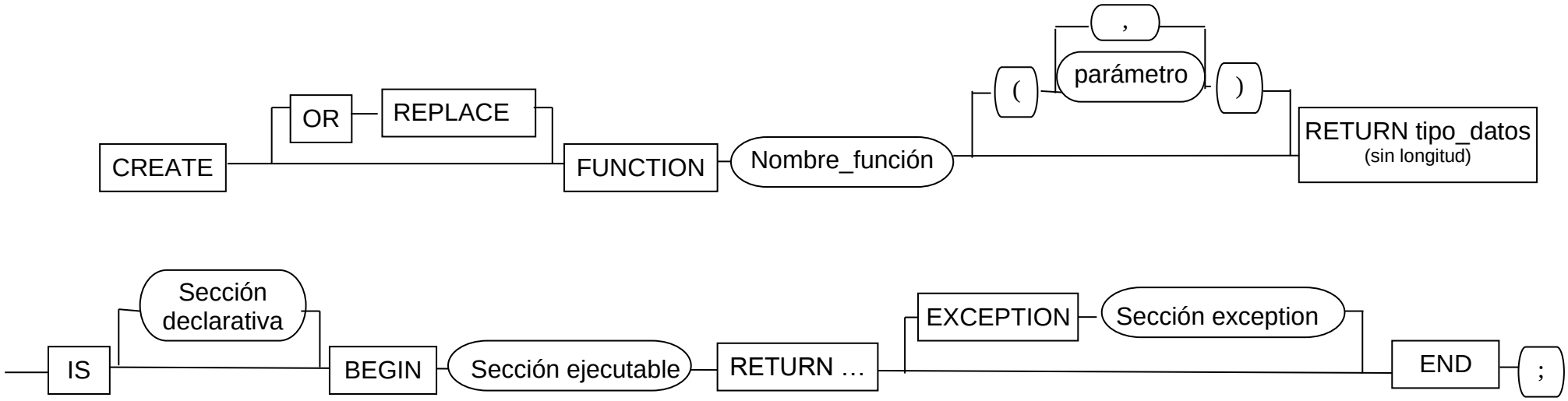
DEFINICIÓN DE FUNCIONES EN PL/SQL: CREATE FUNCTION

SINTAXIS:



Al definir los parámetros se debe poner*: nombre_parámetro tipo_de _parámetro tipo_de_datos(Sin longitud)*
Como tipo de parámetro puede ser:

- IN(por defecto) pudiendo pasarle valores en la llamada a la función. Este valor no se puede modificar. Se le puede asignar un valor por defecto.

  CREATE FUNCTION EJEMPLO(entrada in number default 10) RETURN varchar IS …

- Los tipos OUT e IN OUT no se suelen utilizar en las funciones.

Para borrar una función **DROP FUNCTION nombre_función**

¿CÓMO DEFINIR EXCEPCIONES EN UNA FUNCIÓN?

Como ya vimos en sesiones anteriores, existen dos tipos de excepciones:

- las excepciones predefinidas en Oracle (por ejemplo, no_data_found) y
- las excepciones definidas por el usuario (que tenemos que definir, lanzar y capturar de manera explícita).

¡IMPORTANTE! Las funciones siempre deben devolver "algo", aunque sea "null".

Ejemplo 1:

Crea una función en la que dado un número de habitación devuelva su categoría.

```
create or replace FUNCTION tipo_habitacion  (elnumero habitacion.numero%type) RETURN habitacion.categoria%type
is
auxcategoria habitacion.categoria%type;
begin
select categoria into auxcategoria from habitacion
where numero=elnumero;
return(auxcategoria);
exception
   when no_data_found then
       escribir('No existe ninguna habitación con ese número');
      return null;
end;
```

Para ejecutar directamente una función se puede hacer:

Exec escribir(nom_función(param1)); (aunque el parámetro de entrada de escribir es varchar2, hace la transformación de number y date a varchar2)

SELECT nom_función(param1) FROM dual;

También se puede llamar desde:

- Un procedimiento
- Una sentencia SELECT, tanto como columna en la SELECT como en las condiciones de las cláusulas WHERE y HAVING como en el GROUP BY.
- Una sentencia INSERT dentro de los VALUES.
- Una sentencia UPDATE en la cláusula SET.

Ejemplo:
Dado un nif de un empleado (que suponemos existe en la base de datos, el manejo de excepciones se verá más adelante) obtener la edad que cumple un empleado en el año en curso, suponiendo que en empleados hay una columna fecha_nac)

```
create or replace function edad (elnif empleado.nif%type) return number
is
aux number(2);
begin
select (to_number (to_char(sysdate,'yyyy`)) - to_number (to_char(fecha_nac,'yyyy'))) into aux
from empleados where nif=elnif;
return(aux);
end;
```

- **Otra opción: create or replace function edad(elnif char) return number**

Para los errores de compilación se opera del mismo modo que con los procedimientos

Se puede ejecutar directamente

Select edad(‘11111111A’) from dual;

O bien

select count(*), edad(nif) from empleados group by edad(nif);

O bien dentro de un procedimiento

```
create or replace procedure listadoedad(laedad number) is
cursor emp  is select e.nif nif,nombre, direccion
              from empleado e, empanimacion an
              where e.nif= an.nif;
begin
escribir('EMPLEADOS DE ANIMACIÓN MENORES de '||laedad ||' : ');
for x in emp loop
if edad(x.nif)<laedad then
escribir('Nombre: '||x.nombre||'   direccion   '||x.direccion);
end if;
end loop;
end;
```

exec listadoporedad(25);

Llamada a la función edad

En la definición de los cursores debemos dar nombre a las columnas en las que se utilicen alias de las tablas o las columnas que sean funciones agregadas.
(En el ejemplo no necesariamente hay que llamarla “nif “… podría llamarse empleado o utilizar cualquier otro nombre)