CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Programación de Bases de Datos I

PL/pgSQL

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Conceptos básicos

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Definición de PL/pgSQL

- ***lenguaje procedimental para el sistema de base de datos PostgreSQL.***
- ***permite ampliar la funcionalidad del servidor de base de datos creando objetos de servidor con lógica compleja.***
  - Crear funciones definidas por el usuario, procedimientos almacenados y disparadores.
  - Ampliar el SQL estándar agregando estructuras de control como **if**, **case** y **loop**.
  - Heredar todas las funciones, operadores y tipos definidos por el usuario.

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Unidades léxicas

- ***Delimitadores.***
  - *para representar operaciones entre tipos de datos, delimitar comentarios, etc.*
- ***Identificadores.***
  - *para nombrar elementos de nuestros programas*
- ***Literales.***
  - *para comparar valores o para asignar valores concretos a los identificadores*
- ***Comentarios.***
  - *de una línea se expresaran por medio del delimitador --.*

    ```
    a:=b;  --asignación
    ```

  - *de varias líneas se acotarán por medio de los delimitadores /* y */.*

    ```
    /* Primera línea de comentarios.
    Segunda línea de comentarios. */
    ```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Unidades léxicas

ejercicio

Dada la siguiente línea de código

```
IF A <> B
THEN iguales := FALSE; --No son iguales
```

su descomposición en unidades léxicas sería la siguiente:

- Identificadores: `A`, `B`, `iguales`.
- Identificadores (palabras reservadas): `IF`, `THEN`.
- Delimitadores: `<>`, `:=`, `;`.
- Literales: `FALSE`
- Comentarios: `--No son iguales`.

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

constantes de cadena en bloques anónimos

Para evitar escapar de todas las comillas y barras invertidas, puede usar la cadena delimitada con caracteres dólar

```
do
$$declare
   agentes_count integer;
begin
   select count(*) into agentes_count
   from agentes;
   raise notice 'El número de agentes es: %', agentes_count;
end;$$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

bloques de PL/pgSQL

Sintaxis:

```
[ <<label>> ]
[ declare
    declarations ]
begin
    statements;
  ...
end [ label ];
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

bloques de PL/pgSQL

Ejemplo (bloque anónimo):

```
do $$
<<first_block>>
declare
  oficinas_count integer := 0;
begin
  -- cuenta el número de oficinas
  select count(*)
  into oficinas_count
  from oficinas;
  -- muestra un mensaje
  raise notice 'El número de oficinas es %', oficinas_count;
end first_block $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Variables ~~y constantes~~

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

variables

Sintaxis:

```
variable_name data_type [:= expression];
```

Ejemplo:

```
do $$
declare
  counter integer := 1;
  first_name varchar(50) := 'John';
  last_name varchar(50) := 'Doe';
  payment numeric(11,2) := 20.5;
begin
  raise notice '% % % has been paid % USD',
       counter,
       first_name,
       last_name,
       payment;
end $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

SELECT INTO

Asignar datos de la base de datos a una variable:

```
select select_list into variable_name from table_expression;
```

Ejemplo:

```
do $$
declare
  agentes_count integer;
begin
  -- select el número de agentes de la tabla agentes
  select count(*)
  into agentes_count
  from agentes;

  -- muestra el número de agentes
  raise notice 'El número de agentes es: %', agentes_count;
end;
$$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

asignar tipos de la base de datos

Ejemplo:

```
do $$
declare
  oficina_nombre oficinas.nombre%type;
  oficina_domicilio oficinas.nombre%type;
begin
  -- obtener nombre de la oficina con id 2
  select nombre
  from oficinas
  into oficina_nombre
  where identificador = 2;
  -- muestra el nombre de la oficina
  raise notice 'Nombre de la oficina con id 2: %', oficina_nombre;
end; $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

ROWTYPE

Para almacenar una fila completa en una variable:

```
row_variable table_name%ROWTYPE;
```

Ejemplo:

```
do $$
declare
  agente_seleccionado agentes%rowtype;
begin
-- selecciona al agente con identificador 11
  select * from agentes into agente_seleccionado where identificador = 11;

-- muestra la información del agente
  raise notice 'El nombre del agente es %', agente_seleccionado.nombre;

  raise notice 'La categoría del agente es %', agente_seleccionado.categoria;
end; $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Mensajes

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

Informes de mensajes

Para generar mensajes, utilizaremos la sentencia `raise`:

```
raise nivel formato;
```

Nivel puede ser uno de los siguientes:

debug, log, notice, info, warning o exception.

Ejemplo:

```
do $$
begin
  raise info 'mensaje informativo %', now() ;
  raise log 'mensaje de log %', now();
  raise debug 'mensaje de depuración %', now();
  raise warning 'mensaje de advertencia %', now();
  raise notice 'mensaje de aviso %', now();
end $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES

Centro Integrado de Formación Profesional

Estructuras de control

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

if - then

Sintaxis

```
if condición then
  sentencias;
end if;
```

Ejemplo:

```
do $$
declare
     oficina_seleccionada oficinas%rowtype;
     id_oficina oficinas.identificador%type := 0;
begin

     select * from oficinas
     into oficina_seleccionada
     where identificador = id_oficina;

     if not found then
     raise notice
          'La oficina % no puede ser encontrada.',
          id_oficina;
     end if;
end $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

if - then - else

Sintaxis

```
if condición then
   sentencias;
else
   sentencias_alternativas;
end if;
```

Ejemplo:

```
do $$
declare
     oficina_seleccionada oficinas%rowtype;
     id_oficina oficinas.identificador%type := 1;
begin
     select * from oficinas into oficina_seleccionada
     where identificador = id_oficina;

     if not found then
     raise notice
          'La oficina % no puede ser encontrada.',
          id_oficina;
     else
     raise notice'El nombre de la oficina es %.',
          oficina_seleccionada.nombre;
     end if;
end $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

case

Sintaxis

```
case expresión-buscada
     when expresión_1 [, expresión_2, ...] then
     sentencias-when
[ ... ]
[else
     sentencias-else
]
END case;
```

case

```
do $$
declare
     oficina_seleccionada oficinas%rowtype;
     id_oficina oficinas.identificador%type := 1;
     provincia varchar(20);
begin
     select * from oficinas into oficina_seleccionada where identificador = id_oficina;
     if not found then
     raise notice'La oficina % no puede ser encontrada.', id_oficina;
     else
     case trunc(oficina_seleccionada.codigo_postal::int / 1000)
          when 28 then provincia := 'Madrid';
          when 30 then provincia := 'Murcia';
          when 27 then provincia := 'Jaén';
          when 36 then provincia := 'Granada';
          else
               provincia := 'Desconocida';
     end case;
     raise notice'La oficina con dirección en % está en la provincia de %.',
          oficina_seleccionada.domicilio, provincia;
     end if;
end $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

loop

Sintaxis

```
<<label>>
loop
     sentencias;
     if condición then
          exit;
     end if;
end loop;
```

Ejemplo:

```
do $$
declare
     n integer:= 10; fib integer := 0;
     counter integer := 0 ; i integer := 0 ;
     j integer := 1 ;
begin
     if (n < 1) then
          fib := 0 ;
     end if;
     loop
          exit when counter = n ;
          counter := counter + 1 ;
          select j, i + j into i, j ;
     end loop;
     fib := i;
     raise notice '%', fib;
end; $$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

while

Sintaxis

```
[ <<label>> ]
while condición loop
     sentencias;
end loop;
```

Ejemplo:

```
do $$
declare
     contador integer := 0;
begin
     while contador < 5 loop
          raise notice 'Contador %', contador;
          contador := contador + 1;
     end loop;
end$$;
```

CIFP Carlos III
INFORMÁTICA Y COMUNICACIONES
Centro Integrado de Formación Profesional

for

Sintaxis:

```
[ <<label>> ]
for contador in [ reverse ] from .. to [ by step ] loop
  statements
end loop [ label ];
```

Ejemplo:

```
do $$
begin
  for contador in 1..5 loop
       raise notice 'contador: %', contador;
  end loop;
end; $$;
```