

Programació en BD i extensió procedimental IV

Subprogrammes: procedures i functions



**Institut Rafael
Campalans**

Plaça del Remei, 1
17160 Anglès

Cicle: DAM

Curs: 2022/2023

Mòdul: 02 Bases de Dades

Objectius



- Stored Procedures
- Functions
- Pas de paràmetres
 - IN
 - OUT

Subprogrames: STORED PROCEDURES

Stored Procedures. S'emmagatzema al servidor i s'executa cada vegada que s'invoca. Els stored procedures **NO** retornen un valor.

```
create [or replace] procedure procedure_name(param_list)
  language plpgsql
  as
  $$
  declare
  -- variable declaration
  begin
  -- logic
  end;
  $$
```

CRIDA: call stored_procedure_name(argument_list)

Stored Procedures: Examples

```
create or replace procedure transfer(  
    sender int,    receiver int,    amount dec  
)  
language plpgsql  
as $$  
begin  
    -- subtracting the amount from the sender's account  
    update accounts  
    set balance = balance - amount  
    where id = sender;  
  
    -- adding the amount to the receiver's account  
    update accounts  
    set balance = balance + amount  
    where id = receiver;  
    commit;  
end;$$
```

Subprogrames: FUNCTIONS

Funció. S'emmagatzema al servidor i s'executa cada vegada que s'invoca. Les funcions retornen un valor.

```
create [or replace] function function_name(param_list)
    returns return_type
    language plpgsql
    as
$$
declare
-- variable declaration
begin
    -- logic
end;
$$
```

RETURN amb aquesta sentència retornem la dada des de la funció. Una funció que no retorna res (VOID) seria com un stored procedure

Funcions: Exemple

```
create function get_film_count(len_from int, len_to int)
returns int
language plpgsql
As $$
declare
    film_count integer;
begin
    select count(*)
    into film_count
    from film
    where length between len_from and len_to;

    return film_count;
end;
$$;
```

Crides a subprogrames

Des d'una comanda SELECT (només funcions)

```
Select get_film_count(40,90);
```

```
GET_FILM_COUNT
```

```
-----
```

```
325
```

Des d'un bloc anònim (procediments i funcions)

```
do $$  
begin  
    raise notice '%', get_film_count(40,90);  
end $$;
```

```
NOTICE: 325
```

Subprogrames: Pas de paràmetres

Pas de paràmetres:

IN. Són els paràmetres que en altres llenguatges s'anomenen per valor. El procediment rep una còpia del valor o variable que s'utilitza com a paràmetre al cridar el procediment. Aquests paràmetres poden ser: valors literals (18 per exemple), variables (v_num per exemple) o expressions (com v_num+18). A aquests paràmetres se'ls hi pot assignar un valor per defecte.

Subprogrames: Pas de paràmetres

- Pas de paràmetres (IN Exemple)

```
create or replace function find_film_by_id(p_film_id int)
returns varchar
language plpgsql
as $$

declare
    film_title film.title%type;
begin
    -- find film title by id
    select title into film_title
    from film where film_id = p_film_id;
return film_title;
end;$$
```

Subprogrames: Pas de paràmetres

Pas de paràmetres:

OUT. Relacionats amb el pas per variable o referència d'altres llenguatges. Només poden ser variables i no poden tenir un valor per defecte. Son variables que es poden utilitzar per enviar un valor al procediment i/o perquè el procediment hi guardi algun valor. És a dir, els paràmetres OUT són variables que s'envien al procediment de manera que si en el procediment canvia el seu valor, aquest valor es manté quan el procediment acaba.

Subprogrames: Pas de paràmetres

Pas de paràmetres (OUT Exemple)

```
create or replace function get_film_stat(  
    out min_len int,  
    out max_len int,  
    out avg_len numeric)  
language plpgsql  
as $$  
begin  
  
    select min(length) ,  
           max(length) ,  
           avg(length)::numeric(5,1)  
    into min_len, max_len, avg_len  
    from film;  
  
end;$$
```

Subprogrames: Pas de paràmetres

Pas de paràmetres

INOUT. Són una combinació dels dos anteriors. Es tracta de variables on el valor es pot utilitzar pel procediment que, a més a més, pot guardar-hi un valor. No se'ls hi pot assignar un valor per defecte.

Subprogrames: Pas de paràmetres

Pas de paràmetres (INOUT Exemple)

```
create or replace function swap(  
    inout x int,  
    inout y int  
)  
language plpgsql  
as $$  
begin  
    select x,y into y,x;  
end; $$;
```

Procediments i funcions



- Crear un procediment (funció que no retorna cap valor) que tregui per pantalla “Hello World!”
- Crear una funció CONCATENAR que passant dos cadenes com a paràmetre tregui per pantalla les dues cadenes concatenades. (Paràmetres IN)
- Crear una funció CONCATENAR2 que passant dos cadenes com a paràmetre retorni una cadena amb el resultat de la concatenació
- Crear una funció que et retorni el dia de la setmana passant-li com a paràmetre una data.

Activitat

A03 Procediments emmagatzemats i triggers



- Connectat al Moodle i descarrega't la pràctica "A03 Procediments i triggers".
- Si tens dubtes, revisa els exemples que tens penjat en el moodle.
- D'aquesta pràctica podràs fer la primera part. (Sense triggers)
- Temps 90m

WEBGRAFIA

- SQL Tutorial, W3schools, Setembre 2022, <https://www.w3schools.com/sqL/default.asp>
- PostgreSQL Tutorial from scratch, Setembre 2022, <https://www.postgresqltutorial.com/>
- Exercicis Online de SQL, W3schools, Setembre 2022, https://www.w3schools.com/SQL/sql_exercises.asp
- PostgreSQL Exercices, Practice,Solution, W3resource,Setembre 2022, <https://www.w3resource.com/postgresql-exercises/>
- PostgreSQL Documentation, PostgreSQL, Setembre 2022, <https://www.postgresql.org/docs/>