Programació en BD i extensió procedimental II
Missatges i cursors

Institut Rafael Campalans
Plaça del Remei, 1
17160 Anglès

Cicle: DAM
Curs: 2022/2023
Mòdul: 02 Bases de Dades

Objectius

- Missatges a l'usuari (RAISE)
  - Tipus de missatges
  - Excepcions
- Cursors
  - Declarar
  - Obrir
  - Recorrer
  - Tancar

Missatges a l'usuari: RAISE

- Mitjançant la instrucció **RAISE** podem enviar missatges a l'usuari. Tenim diferents nivells de missatge i en funció de cada nivell (level) es reporten d'una manera o altre.

```
raise level format;
```

- **Level**:
  debug info log warning notice exception (per defecte)

- **Format**: Cadena de caràcters que representa el missatge a mostrar, utilitza el comodí % per substituir el valor dels paràmetres que li passarem, separats per ','. Hi ha d'haver el mateix nombre de % que paràmetres.

Missatges a l'usuari: RAISE

- **Exemple:**

```
do $$
begin
   raise info 'information message %', now() ;
   raise log 'log message %', now();
   raise debug 'debug message %', now();
   raise warning 'warning message %', now();
   raise notice 'notice message %', now();
end $$;
```

- **Sortida**

```
info:  information message 2015-09-10 21:17:39.398+07
warning:  warning message 2015-09-10 21:17:39.398+07
notice:  notice message 2015-09-10 21:17:39.398+07
```

Missatges d'excepció: RAISE EXCEPTION

Instrucció RAISE EXCEPTION:

- Quan volem enviar un missatge d'error treballarem amb el nivell Exception (per defecte).
- A més podem afegir informació addicional amb la comanda using.

```
using option = expression
```

Option:

message : Missatge d'error
hint: Informació resumida per clarificar l'error
detail: Informació detallada de l'error.
errcode: Informació d'identificació de l'error.
Es pot expressar per nom de l'error o codi d'error (SQLSTATE)

Missatges d'excepció: RAISE EXCEPTION

Instrucció RAISE EXCEPTION

Exemple:

```
do $$
declare
  email varchar(255) := 'info@postgresqltutorial.com';
begin
  -- check email for duplicate
  -- ...
  -- report duplicate email
  raise exception 'duplicate email: %', email
        using hint = 'check the email again';
end $$;
```

```
[Err] ERROR:  Duplicate email: info@postgresqltutorial.com
HINT:  Check the email again
```

https://www.postgresql.org/docs/current/errcodes-appendix.html

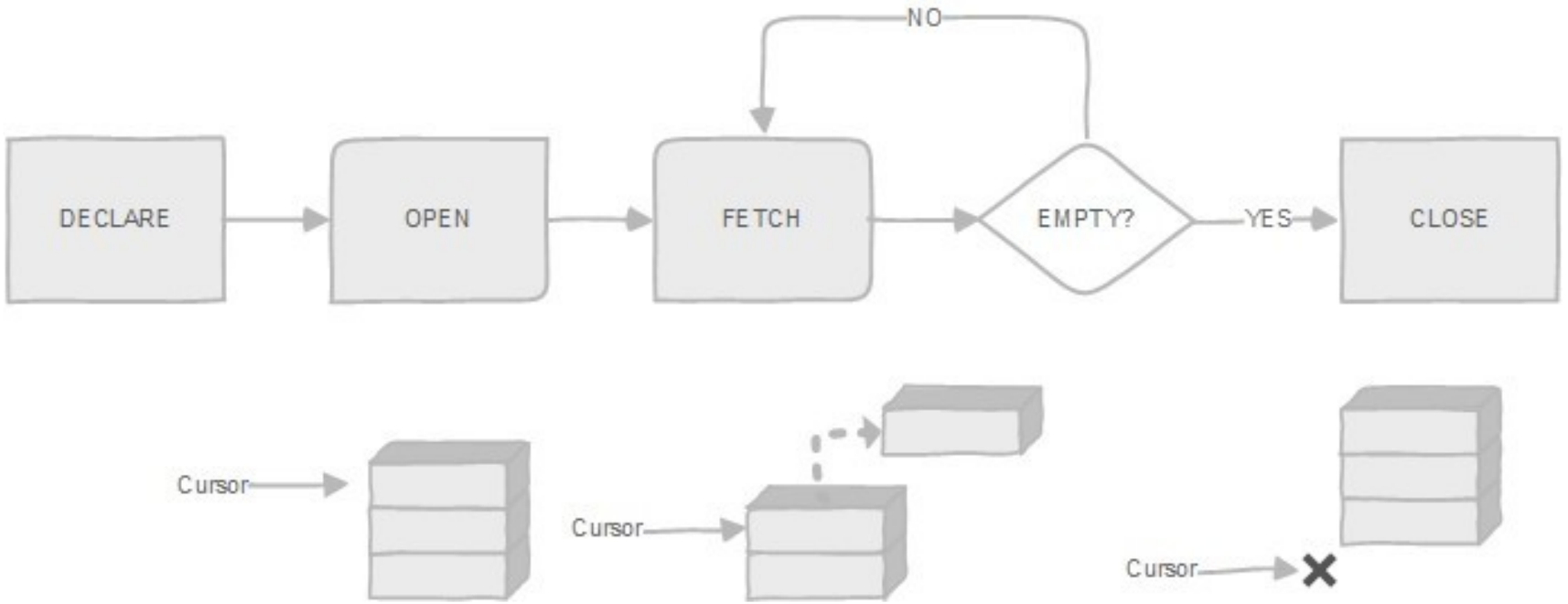
Cursors

- Els **cursors** representen consultes SELECT sql que retornen més d'una fila i que permeten l'accés a cadascun dels registres. Això significa que el cursor sempre té un **punter** a una de les files del SELECT que representa el cursor.

- Es processen amb **quatre passes**:

  1. Declarar el cursor
  2. Obrir el cursor. Després d'obrir el cursor, aquest apuntarà a la primera fila (si n'hi ha).
  3. Avançar el cursor. La instrucció **FETCH** permet recórrer el cursor registre a registre fins que el punter arribi al final.
  4. Tancar el cursor.

Cursors



Cursors: Declarar

- **Pas 1. Declarar el cursor:**

```
cursor_name [ [no] scroll ] cursor
[( name datatype, name data type, ...)] for query;
```

```
declare
    cur_films  cursor for
        select *
        from film;
    cur_films2 cursor (year integer) for
        select *
        from film
        where release_year = year;
```

- (Veure'm més endavant que també podem utilitzar el tipus record)

Cursors: Obrir

- **Pas 2. Obrir el cursor:**

```
open cursor_variable[ (name:=value,name:=value,...)];
```

1. Reservar memòria suficient pel cursor.
2. Executar la sentència SELECT que defineix el cursor.
3. Situar el punter a la primera fila.

- Si no hi ha registres per mostrar PostgreSQL no retorna cap error.

Cursors: Obrir

- **Pas 3. Avançar el cursor:**

```
fetch [ direction { from | in } ] cursor_variable
into target_variable;
```

Direction: Indica la direcció d'avanç, pot ser:
NEXT, LAST, PRIOR, FIRST, ABSOLUTE cont, RELATIVE cont, FORWARD, BACKWARD

Cursors: Recorregut

- **Pas 3. Avançar el cursor:**

- Podem moure el cursor però sense retornar les dades de la fila amb la sentència MOVE que admet els mateixos paràmetres que FETCH.

```
fetch cur_films into row_film;
fetch last from row_film into title, release_year;

move cur_films;
move last from cur_films;
move relative -1 from cur_films;
move forward 3 from cur_films;
```

Cursors: Tancar

Pas 4. Tancar el cursor.

```
close cursor_variable;
```

- Quan es tanca el cursor s'allibera la memòria que ocupa i s'impedeix que s'utilitzi.

Cursors: Exemple

```
do $$
Declare
    titles text default '';
    rec_film   record;
    cur_films cursor(p_year integer)
    for select title, release_year
        from film
        where release_year = p_year;
begin
   -- open the cursor
   open cur_films(p_year:=2006);

   loop
    -- fetch row into the film
      fetch cur_films into rec_film;
    -- exit when no more row to fetch
      exit when not found;

    -- build the output
      if rec_film.title like '%FUL%' then
         titles := titles || ',' || rec_film.title || ':' || rec_film.release_year;
      end if;
   end loop;

   -- close the cursor
   close cur_films;

   raise notice '%', titles;
end $$;
```

Cursors: Variables tipus Record

Variables tipus Registre:

- Quan volem assignar un conjunt de dades d'una fila retornada per una consulta podem utilitzar el tipus record.

```
variable_name record;
```

- Aquestes variables no tenen una estructura predeterminada sinó que vindrà donada per el resultat de la consulta on la utilitzem.

- Recordeu que també tenim la possibilitat de declarar variables tipus %ROWTYPE.

Cursors amb registres exemple

```
do $$
declare
        rec record;
begin
        -- select the film
        select film_id, title, length
        into rec
        from film
        where film_id = 200;

        raise notice '% % %', rec.film_id, rec.title,
                              rec.length;
end;
$$
```

Cursors: Recorregut amb FOR

FOR per treballar amb cursors:

- És la forma més normal de recórrer totes les files d'un cursor. És un bucle FOR que s'encarrega de fer tres tasques:
  - Obre el cursor (fa l'OPEN abans de començar el bucle).
  - Recorre totes les files del cursor (a cada iteració es genera un FETCH implícit) i emmagatzema el contingut de cada fila en una variable de registre (que no cal declarar a la zona DECLARE).
  - Tanca el cursor (quan acaba el FOR).

```
[ <<label>> ]
FOR recordvar IN bound_cursorvar
[ ( [ argument_name := ] argument_value [, ...] ) ] LOOP
    statements
END LOOP [ label ];
```

Cursors: Exemple recorregut

- Bucle FOR per treballar amb cursors:

Exemple:

```
do $$
declare
cur_films cursor(p_year integer)  for select title,
release_year
from film
where release_year = p_year;
titles text := '';
begin
-- open and fetch cursor
for rec_film in cur_films(p_year:=2006) loop
-- build the output
if rec_film.title like '%FUL%' then
titles := titles || ',' || rec_film.title || ':' ||
rec_film.release_year;
end if;  end loop;
raise notice '%', titles;
end $$;
```

Cursors

- Bucle FOR per treballar amb cursors, escrivint la comanda select en el bucle:

Exemple:

```
do $$
declare
   titles text := '';
   rec_film record;
   p_year int := 2006;
begin
   -- open the cursor
   for rec_film in (select title, release_year from film
                    where release_year = p_year)  loop
    -- build the output
      if rec_film.title like '%FUL%' then
         titles := titles || ',' || rec_film.title || ':'
                   || rec_film.release_year;
      end if;
   end loop;
   raise notice '%', titles;
end $$;
```

Cursors

- Cursors per actualitzar registres:
- A vegades un cursor ens podrà fer servei per actualitzar els seus registres. Una vegada el cursor està obert podem actualitzar o eliminar el registre apuntat:

```
update table_name
set column = value, ...
where current of cursor_variable;

delete from table_name
where current of cursor_variable;
```

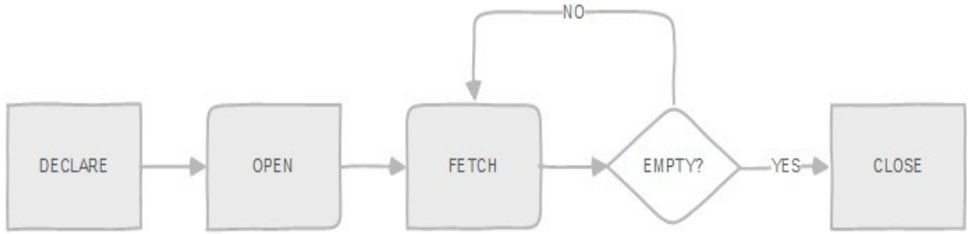
```
update film
set release_year = p_year
where current of cur_films;
```

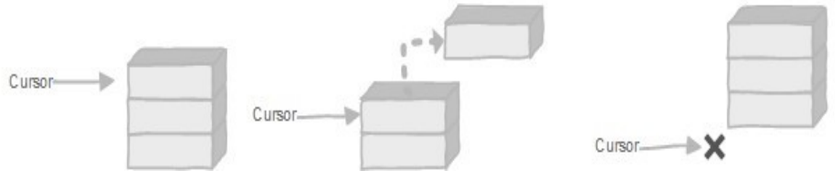
Resum

Missatges

```
raise level format;
```

Cursors





Activitat

A02 Blocs anònims i cursors

- Connectat al Moodle i descarrega't la pràctica "A02 PL/pgSQL - Blocs anònims i cursors".
- Si tens dubtes, revisa els exemples que tens penjat en el moodle.
- Temps 90m

WEBGRAFIA

- PostgreSQL Config Logging, PostgreSQL, Setembre 2022, https://www.postgresql.org/docs/current/runtime-config-logging.html
- PostgreSQL Errors and messages, PostgreSQL, Setembre 2022, http://www.postgresql.org/docs/current/static/plpgsql-errors-and-messages.html
- PostgreSQL Cursors, PostgreSQL, Setembre 2022, https://www.postgresql.org/docs/current/plpgsql-cursors.html

- 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/