



Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT10. SQL MODO PROGRAMACIÓN

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz



CONTENIDOS

- Introducción. Conceptos generales
- Variables, operadores, expresiones
- Bloques instrucciones de código
- Estructuras de control.
- Procedimientos y funciones almacenados.
- Cursores
- Manejo de errores



CURSORES

- Dentro de las funciones y procedimientos podemos trabajar con instrucciones que pertenecen al DML.
- Cuando dichas instrucciones involucran más de una fila (varios registros) necesitamos utilizar una variable especial para almacenar la información. Esta variable se llama CURSOR.
- Con el cursor podemos recorrer la información registro por registro.



SINTAXIS

- Un cursor es una zona de memoria que contiene un conjunto de filas resultantes de una sentencia SQL con la ventaja de que podremos recorrer, visualizar y manipular una a una cada una de esas filas.

```
DECLARE nombre_del_cursor CURSOR FOR sentencia_select;
```

Los cursores se declaran en los procedimientos u otros programas almacenados después de la declaración de variables, no hacerlo así producirá una situación de error.

CURSORES

¿Qué podemos hacer con un cursor?

- DECLARE nombreDelCursor CURSOR FOR SELECT ...;
- OPEN nombreDelCursor;
- FETCH nombreDelCursor INTO unaVariable [, otraVariable];
- CLOSE nombreDelCursor;

FETCH = Extraer datos del cursor

Ejemplo:

```
DECLARE miCursor CURSOR FOR SELECT campo1, campo2,  
    campo3 FROM unaTabla;
```

Los cursores son de sólo-lectura, no se puede hacer UPDATE de un cursor. Se leen línea a línea y sólo en un sentido (siempre para delante, no se puede volver a una fila que ya se ha leído)



EJEMPLO (A)

```
1  /*Ejemplo cursor */
2  • use curso;
3  • DROP PROCEDURE IF EXISTS cursor1;
4  DELIMITER //
5  • CREATE PROCEDURE cursor1 ()
6  BEGIN
7      -- 1ª Declaración variables
8      declare dnombre_aux, localidad_aux varchar(25);
9
10     -- 2ª Declaración cursores
11     declare cursor_dptos cursor for SELECT DNOMBRE, LOCALIDAD FROM DEPARTAMENTOS ;
12
13     -- Cuerpo
14     Open cursor_dptos; -- Abrir cursor
15     recorrer_dptos: loop -- Recorrerlo
16         fetch cursor_dptos into dnombre_aux, localidad_aux;
17
18         -- Tratamiento del registro recuperado. En nuestro caso mostrarlo
19         select dnombre_aux, localidad_aux;
20     end loop recorrer_dptos;
21     close cursor_dptos; -- Cerrar cursor
22 END //
23 DELIMITER ;
24 • call cursor1();
```

Cuando llega a la última fila está vacío y retorna un error

Result Grid Filter Rows: Export: Wrap Cell Content:			
	dnombre_aux	localidad_aux	
▶	MARKETING	BILBAO	✓ 102 11:32:05 call cursor1() 1 row(s) returned
			✓ 103 11:32:05 call cursor1() 1 row(s) returned
			✓ 104 11:32:05 call cursor1() 1 row(s) returned
			✗ 105 11:32:05 call cursor1() Error Code: 1329 No data - zero rows fetched, selected, or processed
Result 6 Result 7 Result 8 Result 9 Result 10 x			

TRATAMIENTO ERRORES

- Para tratar este error, el intento de recuperar una fila habiendo llegado ya al final del cursor, definiremos un manejador de error

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET v_ultima_fila=1;
```



EJ (B)

```
5 • CREATE PROCEDURE cursor2 ()
6 BEGIN
7     -- 1ª Declaración variables
8     declare dnombre_aux, localidad_aux varchar(25);
9     declare ultima_fila int default 0;
10    -- 2ª Declaración cursores
11    declare cursor_dptos cursor for SELECT DNOMBRE, LOCALIDAD FROM DEPARTAMENTOS ;
12
13    -- 3ª Declaración de errores
14    declare continue handler for not found set ultima_fila=1;
15    -- Cuerpo
16    Open cursor_dptos; -- Abrir cursor
17    recorrer_dptos: loop -- Recorrerlo
18        fetch cursor_dptos into dnombre_aux, localidad_aux;
19        if ultima_fila=1 then -- Aseguramos que no llegamos a la última fila
20            leave recorrer_dptos;
21        end if;
22        -- Tratamiento del registro recuperado. En nuestro caso mostrarlo
23        select dnombre_aux, localidad_aux;
24    end loop recorrer_dptos;
25    close cursor_dptos; -- Cerrar cursor
26 END //
27 DELIMITER ;
28 • call cursor2();
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
dnombre_aux	localidad_aux		
MARKETING	BILBAO		

Ya no tenemos error

Result 10	Result 11	Result 12	Result 13	Result 14 x
Output				
Action Output				
#	Time	Action		
✓ 109	11:39:09	call cursor2()		
✓ 110	11:39:09	call cursor2()		
✓ 111	11:39:09	call cursor2()		
✓ 112	11:39:09	call cursor2()		

Estamos declarando un manejador de error para el error NOT FOUND, que es el que se produce cuando se intenta recuperar una fila habiendo llegado al final del cursor.

Se utiliza una variable (v_ultima_fila) que tomará el valor 1 en el momento que se produzca el error.

Dentro de nuestro programa podremos preguntar por su valor.

Cuando se produzca la situación de error, la ejecución del programa continuará (debido a la cláusula CONTINUE) y no finalizará en el punto donde se encuentra el error. En el momento en que la instrucción (FETCH ... INTO) falle por no poder leer ninguna fila del cursor, la variable v_ultima_fila tomará el valor 1 y la ejecución del procedimiento continuará por la línea siguiente al error para abandonar el bucle



EJEMPLO

```
1  /*Ejemplo cursor que inserta todos los departamentos en otra tabla*/
2  ● use curso;
3  ● DROP PROCEDURE IF EXISTS cursor3;
4  DELIMITER //
5  ● CREATE PROCEDURE cursor3 ()
6  BEGIN
7      -- 1ª Declaración variables
8      declare dnombre_aux, localidad_aux varchar(25);
9      declare ultima_fila int default 0;
10     -- 2ª Declaración cursores
11     declare c_todos cursor for SELECT DNOMBRE, LOCALIDAD FROM DEPARTAMENTOS;
12     -- 3ª Declaración de errores
13     declare continue handler for not found set ultima_fila=1;
14     -- Cuerpo
15     DROP TABLE IF exists dptosMadrid;
16     create table dptosMadrid(
17         id int NOT NULL AUTO_INCREMENT primary KEY,
18         dnombre varchar(25),
19         localidad varchar(45)
20     );
21     Open c_todos; -- Abrir cursor
22     recorrer_dptos: loop -- Recorrerlo
23         fetch c_todos into dnombre_aux, localidad_aux;
24         if ultima_fila=1 then -- Aseguramos que no llegamos a la última fila
25             leave recorrer_dptos;
26         end if;
27         -- Tratamiento del registro recuperado. En nuestro caso mostrarlo
28         insert into dptosMadrid (DNOMBRE, LOCALIDAD)
29         select dnombre_aux, localidad_aux;
30     end loop recorrer_dptos;
31     close c_todos; -- Cerrar cursor
32 END //
33 DELIMITER ;
34 ● call cursor3();
35 ● SELECT * FROM dptosMadrid;
```

RECUERDA: El orden de las declaraciones es importante.

- 1) Declarar variables
- 2) Declarar cursores
- 3) Declarar manejadores

Result Grid			
Filter Rows:			
Edit:			
Export/Import:			
Wrap Cell Content:			
	id	dnombre	localidad
▶	1	CONTABILIDAD	BARCELONA
	2	INVESTIGACION	VALENCIA
	3	VENTAS	MADRID
	4	PRODUCCION	SEVILLA
	5	MARKETING	BILBAO
✱	NULL	NULL	NULL



CURSORES ANIDADOS

- Ejemplo, en el que de la bd curso se desea listar los departamentos y total de empleados

```
mysql> call anidados();
+-----+
| resultado |
+-----+
| Departamento 10 - CONTABILIDAD - BARCELONA - Total empleados: 9 |
+-----+
1 row in set (0.01 sec)

+-----+
| resultado |
+-----+
| Departamento 20 - INVESTIGACION - VALENCIA - Total empleados: 9 |
+-----+
1 row in set (0.01 sec)

+-----+
| resultado |
+-----+
| Departamento 30 - VENTAS - MADRID - Total empleados: 9 |
+-----+
1 row in set (0.01 sec)

+-----+
| resultado |
+-----+
| Departamento 40 - PRODUCCION - SEVILLA - Total empleados: 9 |
+-----+
1 row in set (0.02 sec)

+-----+
| resultado |
+-----+
| Departamento 50 - MARKETING - BILBAO - Total empleados: 9 |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)
```

CURSORES ANIDADOS

- Ejemplo, en el que de la bd curso se desea listar los departamentos y luego todos los empleados para ese departamento

MySQL 8.0 Command Line Client

```
+-----+
| NombreDepartamento |
+-----+
| Departamento 50 - MARKETING |
+-----+
1 row in set (0.39 sec)

+-----+
| Empleados |
+-----+
| Empleado: 7876 - GIL |
+-----+
1 row in set (0.40 sec)

+-----+
| NombreDepartamento |
+-----+
| Departamento 50 - MARKETING |
+-----+
1 row in set (0.40 sec)

+-----+
| Empleados |
+-----+
| Empleado: 7900 - JIMENEZ |
+-----+
1 row in set (0.41 sec)

+-----+
| NombreDepartamento |
+-----+
| Departamento 50 - MARKETING |
+-----+
1 row in set (0.41 sec)

+-----+
| Total |
+-----+
| Total empleados: 9 |
+-----+
1 row in set (0.41 sec)
```





Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT10. SQL MODO PROGRAMACIÓN

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

