

Cursores y disparadores

Cursores

- Los cursores son unas potentes herramientas que permiten encapsular consultas, albergarlas en memoria y acceder a ellas en cualquier momento.
- Existen cuatro operaciones básicas relacionadas con cursores: declaración, abertura (OPEN), recogida de información (FETCH), cierre del cursor (CLOSE).

Declaración:

```
DECLARE nombre_cursor CURSOR FOR consulta
```

- Deben ser declarados después de las variables y antes del resto de sentencias.

Abertura:

```
OPEN nombre_cursor;
```

- Es en este momento cuando ejecutamos la consulta.

Recogida de información o extracción:

```
FETCH nombre_cursor INTO nombre_var [, nombre_var2...]
```

- Con esta instrucción recogemos el siguiente registro de la consulta asociada al cursor_nombre y almacenamos el valor de cada campo obtenido en una variable. Por tanto, debe coincidir número de variables con número de campos que devuelve la consulta.

Cierre

```
CLOSE nombre_cursor;
```

Problemas Cursores

- Escribe un procedimiento que imprima por pantalla todos los usuarios del sistema pero utilizando un variable de cursor.
- Reescribe el procedimiento anterior pero utiliza un contador para evitar el error de "No data -zero rows fetched"
- Reescribe el procedimiento anterior para que muestre no solo el usuario sino también el 'Host' asociado.
- Echa un ojo al ejemplo del manual 13.6.6 y trata de averiguar qué realiza.
- Escribe un procedimiento que detecte si hay clientes con el mismo nombre en empresa.CLIENTE y videoclub.CLIENT.

Disparadores o *Triggers*

- Un disparador es una rutina asociada a una tabla que se activa cuando un determinado evento ocurre en esa tabla.

```
CREATE
```

```
[DEFINER = { user | CURRENT_USER }]
```

```
TRIGGER trigger_name trigger_time trigger_event
```

```
ON tbl_name FOR EACH ROW trigger_body
```

- Requiere el TRIGGER PRIVILEGE (y puede requerir el de SUPER dependiendo del valor del DEFINER)

- trigger_time= BEFORE | AFTER
- trigger_event= indica el tipo de instrucción que activa el disparador. En principio se contemplan los siguientes eventos: INSERT, UPDATE, DELETE
- No se puede disponer de dos disparadores para la misma tabla que se activen para el mismo evento y en el mismo momento.
- atrigger_body= composición de instrucciones BEGIN..END.
- Para referirnos a los registros de la tabla a la que el disparador está asociado tenemos la posibilidad de utilizar los alias NEW.Nombre_columna y OLD.Nombre_columna.

Problemas Disparadores

- Escribe un disparador que muestre un mensaje cualquiera al realizar una inserción en la tabla empresa.CLIENTE.
- Escribe un disparador que contabilice -utilizando una tabla aparte- el número de modificaciones que sufre la tabla comanda.
- Escribe un disparador que acumule en una variable de usuario la suma de los "totales" de la tabla comanda cada vez que haya una inserción.
- Escribe un disparador que se active al eliminar un registro en la tabla COMANDA y que reste el valor de TOTAL del registro eliminado a la variable de usuario creada en el ejercicio anterior.
- Escribe un disparador que, cuando se produzca una actualización en COMANDA, impida que el campo COM_TIPUS varíe.