Bases de Datos

Unidad 8:

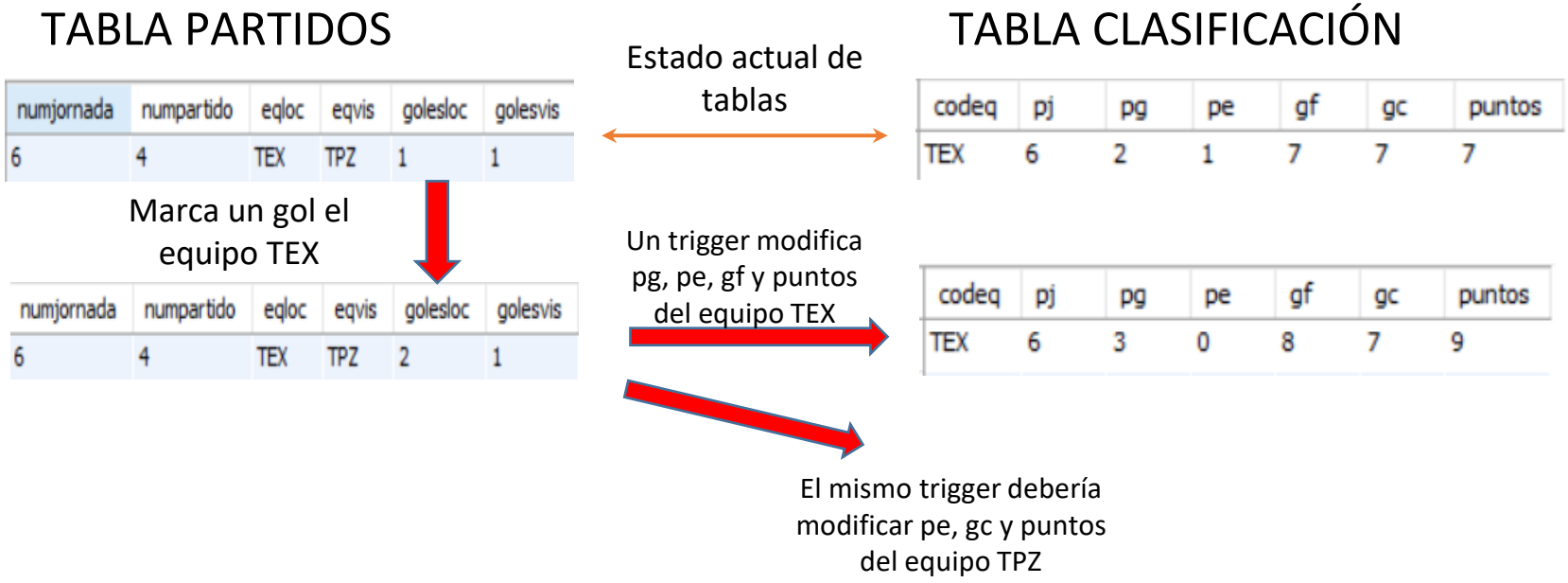
Programación de bases de datos

Sesión 6

5.- Triggers

*Un **trigger o disparador** es una rutina (conjunto de sentencias) que **se lanza a ejecución automáticamente cuando** se produce un **evento** de actualización de datos sobre una tabla (INSERT, UPDATE, DELETE).*

Un ejemplo: ¿Qué se debe desencadenar en una clasificación de una liga de fútbol cuando se <u>modifica</u> el resultado de un partido?.



SINTAXIS PARA CREAR UN TRIGGER

```
CREATE TRIGGER nombreTrigger  momento_disparo   evento    ON nombreTabla
FOR EACH ROW
BEGIN
        sentencias;
END
```

- ***Evento puede ser INSERT, UPDATE, DELETE. Es una acción realizada sobre una tabla que va a desencadenar la realización automática de otras acciones sobre otras tablas.***
- ***Momento_disparo*** *especifica si las sentencias se ejecutan antes que el evento que lanza al trigger* ***(BEFORE)*** *o después* ***(AFTER).*** *En muchos casos puede dar igual usar BEFORE o AFTER.* ***Se debe usar BEFORE si trata de validarse que se puede efectuar el evento. Por ejemplo, al insertar un contrato de alquiler de un coche, debería lanzarse un trigger que comprobase si está disponible para alquilar, si no lo está, el trigger debería abortar el evento.***
- ***FOR EACH ROW*** *indica que el trigger se lanza por cada fila afectada por el evento.*

LOS OPERADORES NEW y OLD

• *Dentro de las sentencias que se ejecutarán al dispararse el trigger, se pueden usar los operadores* ***OLD y NEW.*** *Estos operadores sirven para hacer referencia a las columnas de las filas afectadas por un evento dentro del trigger.*

• ***El operador NEW*** *sirve para hacer referencia al nuevo valor de una columna sobre la que se produce un evento y se usa como* ***NEW.nombreColumna****.*

• *El* ***operador OLD*** *sirve para hacer referencia al anterior valor de una columna sobre la que se produce un evento y se usa en la forma* ***OLD.nombreColumna****.*

LOS OPERADORES NEW y OLD

DML → / Pseudo Records ↓	Insert	Update	Delete
New	✓	✓	✗
Old	✗	✓	✓

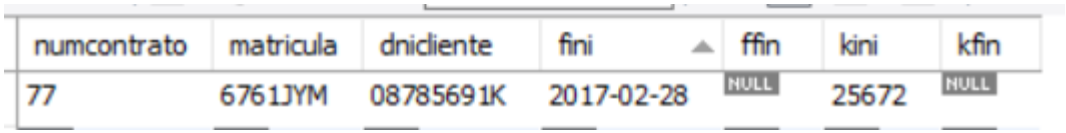
Programación de triggers. Restricciones

- Nunca puede haber dos triggers para responder a un mismo evento sobre una misma tabla en el mismo momento de disparo (veremos después que es esto).

- No se permite usar sentencias que devuelvan filas de resultados. Si que se permiten sentencias SELECT que devuelvan una fila y carguen lo devuelto en variables (SELECT … INTO … FROM)

LOS OPERADORES NEW y OLD

Supongamos que tenemos este contrato y que tenemos un trigger sobre los eventos UPDATE de contratos.



numcontrato	matricula	dnicliente	fini	ffin	kini	kfin
77	6761JYM	08785691K	2017-02-28	NULL	25672	NULL

Y que ejecutamos: ***UPDATE contratos SET fini=adddate(fini,interval 1 week), ffin='2017-05-23', kfin=27200 where numcontrato=77;***

Estos serían los valores ***OLD y NEW*** *de* ***contratos*** *mientras se está ejecutando el trigger:*

COLUMNA	OLD	NEW
numcontrato	77	77
matricula	6761JYM	6761JYM
dnicliente	08785691K	08785691K
fini	2017-02-28	2017-03-07
ffin	null	2017-05-23
kfin	null	27200

EJEMPLO 1: Realizar un trigger que, tras añadir un nuevo contrato de alquiler de un coche, actualiza el estado de alquilado de ese coche.

```
CREATE TRIGGER alquilar AFTER INSERT ON contratos FOR EACH ROW
BEGIN
  UPDATE automoviles SET alquilado=true WHERE matricula=NEW.matricula;

END
```

NEW.matricula hace referencia a la nueva matricula afectada por el evento (INSERTAR en la tabla CONTRATOS). Por tanto hace referencia a la matricula insertada en el nuevo contrato.

El momento de disparo podría ser también BEFORE.

PRUEBA: Inserta un nuevo contrato en contratos y comprueba que el estado de alquilado del coche cambia.

EJEMPLO 2: Modifica el trigger anterior para que se asigne a los kilómetros iniciales del contrato insertado los kilómetros que tiene el coche contratado.

```
CREATE TRIGGER alquilar BEFORE INSERT ON contratos FOR EACH ROW
BEGIN
   DECLARE k INT;
-- obtenemos los kilómetros del coche que se va a contratar
   SELECT kilometros INTO k FROM automoviles WHERE matricula=NEW.matricula;
-- asignamos los kilómetros al nuevo valor que se va a insertar en contratos
   SET NEW.kini=k;
   UPDATE automoviles SET alquilado=true WHERE matricula=NEW.matricula;
END
```

El momento de ejecución del trigger tiene que ser BEFORE (antes que se modifique). Si pusiéramos AFTER, no tendría efecto **SET NEW.kini=k**, puesto que ya se habría insertado el contrato.

PRUEBA: Inserta un nuevo contrato en contratos. ¿Cambian los kilómetros iniciales del coche?

EJEMPLO 3: Modifica el trigger anterior para que, además de lo que realizaba, compruebe si el coche a contratar se puede alquilar, es decir, no está alquilado. Si está alquilado, se debe evitar que se haga el contrato.

```
CREATE TRIGGER alquilar BEFORE INSERT ON contratos FOR EACH ROW
BEGIN
DECLARE k INT;
DECLARE a BOOLEAN;
SELECT kilometros, alquilado INTO k,a FROM automoviles WHERE matricula=new.matricula;
 IF a=true THEN
            SET NEW.matricula=null;
ELSE
            SET NEW.kini=k;
            UPDATE automoviles SET alquilado=true WHERE matricula=NEW.matricula;
END IF;
END
```

Al poner **new.matricula** a **null**, no se actualiza ya que no se admiten nulos en la columna matrícula de contratos. Entonces, se produce un error de ejecución y se aborta el proceso, se sale del trigger.

EJEMPLO 4: Realiza un trigger para que, al hacer la modificación de un contrato correspondiente a la finalización de un contrato, se establezca que el coche pasa a estar disponible y que los kilómetros del coche sean los kilómetros finales del coche en el contrato.

```
CREATE TRIGGER entregar AFTER UPDATE ON contratos FOR EACH ROW
BEGIN
DECLARE k INT;
DECLARE a BOOLEAN;
IF OLD.ffin IS NULL AND NEW.ffin IS NOT NULL THEN
      UPDATE automoviles SET alquilado=false WHERE matricula=NEW.matricula;
      IF NEW.kfin IS NOT NULL THEN
            UPDATE automoviles SET kilometros=NEW.kfin WHERE
matricula=NEW.matricula;
     END IF;
END IF;
END
```

Para comprobar si es una modificación por una entrega, se verifica si la fecha final contenía null y se ha cargado un valor de fecha en el contrato. Sólo se asignan kilómetros al coche cuando se haya cargado un nuevo valor en el contrato.

EJEMPLO 5: Suponiendo que se tiene una tabla **auditoriaCLIENTES**, con las columnas usuario, dia, hora, instrucción, dni, realiza un trigger tal que, al eliminar algún cliente en la tabla clientes, añada una fila en la tabla **auditoriaCLIENTES** indicando quien y cuando hizo la eliminación y que dni de cliente se eliminó. En la columna dni se almacena el dni del cliente eliminado. En la columna instrucción se carga la instrucción auditada (INSERT, UPDATE o DELETE).

```
CREATE TRIGGER auditarC AFTER DELETE ON clientes FOR EACH ROW
BEGIN
        INSERT INTO auditoriaclientes (usuario, dia, hora, instruccion, dni)
        VALUES (current_user(), curdate(), curtime(), 'delete', OLD.dni);
END
```