

Práctica de disparadores

Juan Carlos Llamas Núñez 3ºDG

En este documento se presentan las capturas del código para facilitar la legibilidad, así como las capturas del contenido de las tablas. Al final del documento se añade el texto plano del archivo .sql. En relación a las encuestas, he contestado ambas, pero me di cuenta tras hacer la entrega de que se me había olvidado proponer otra forma de mejor DES: conversión automática de tipos (en particular de entero a real sería muy útil).

1a

```
DROP TABLE contiene;
DROP TABLE pedidos;
DROP TABLE auditoría;
DROP VIEW vista_pedidos;

CREATE TABLE pedidos(código CHAR(6) NOT NULL, fecha CHAR(10) NOT NULL,
importe NUMBER(6,2) DEFAULT 0, cliente CHAR(20) NOT NULL, notas CHAR(1024),
especial CHAR(1) NOT NULL, CHECK (especial in('S','N')), CHECK (importe>=0),
PRIMARY KEY(código));

CREATE VIEW vista_pedidos AS SELECT * FROM pedidos;

CREATE TABLE contiene(pedido CHAR(6) REFERENCES pedidos(código), plato
CHAR(20) NOT NULL, precio NUMBER(6,2) NOT NULL, unidades NUMBER(2,0) NOT NULL,
CHECK (precio>=0), CHECK (unidades>0), PRIMARY KEY(pedido,plato));

CREATE TABLE auditoría(operación CHAR(6) NOT NULL, tabla CHAR(50) NOT NULL,
fecha CHAR(10) NOT NULL, hora CHAR(8) NOT NULL,
CHECK (operación in('DELETE','INSERT','UPDATE')));
--Esta tabla no contiene Primary key porque puedes hacer operaciones
--simultaneas del mismo tipo en una misma tabla y en un mismo instante
```

1b

```
CREATE OR REPLACE TRIGGER tr_pedidos
AFTER INSERT OR DELETE OR UPDATE ON pedidos
BEGIN
    IF DELETING THEN
        INSERT INTO auditoría VALUES('DELETE','pedidos',
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));
    ELSIF INSERTING THEN
        INSERT INTO auditoría VALUES('INSERT','pedidos',
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));
    ELSIF UPDATING THEN
        INSERT INTO auditoría VALUES('UPDATE','pedidos',
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));
    END IF;
END tr_pedidos;
/

BEGIN
    INSERT INTO pedidos VALUES('123456','11/12/2020',0,'Juan Carlos',
        'Entrar por la chimenea','S');
    UPDATE pedidos SET importe=importe+1 WHERE código='123456';
    DELETE FROM pedidos WHERE código='123456';
END;
/
```

OPERACIÓN		TABLA	FECHA	HORA
1	INSERT	pedidos	18/12/2020	04:44:21
2	UPDATE	pedidos	18/12/2020	04:44:21
3	DELETE	pedidos	18/12/2020	04:44:21

2 (Se muestra la evolución de las tablas contiene y pedidos tras ejecutar cada instrucción)

```
CREATE OR REPLACE TRIGGER tr_contiene
AFTER INSERT OR DELETE OR UPDATE ON contiene
FOR EACH ROW
BEGIN
    IF DELETING THEN
        UPDATE pedidos SET importe= importe - :OLD.precio*:OLD.unidades
        WHERE :OLD.pedido=código;
    ELSIF INSERTING THEN
        UPDATE pedidos SET importe=:NEW.precio*:NEW.unidades
        WHERE :NEW.pedido=código;
    ELSIF UPDATING THEN
        UPDATE pedidos SET importe=importe-:OLD.precio*:OLD.unidades +
        :NEW.precio*:NEW.unidades WHERE :OLD.pedido=código;
    END IF;
END tr_contiene;
/

BEGIN
    INSERT INTO pedidos VALUES('123456','11/12/2020',0,'Juan Carlos',
    'Entrar por la chimenea','S');
    INSERT INTO contiene VALUES('123456','Tortilla',20,2);
    UPDATE contiene SET precio=precio+5, unidades = unidades + 6
    WHERE pedido='123456'and plato='Tortilla';
    DELETE FROM contiene WHERE pedido='123456';
    DELETE FROM pedidos WHERE código='123456';
END;
```

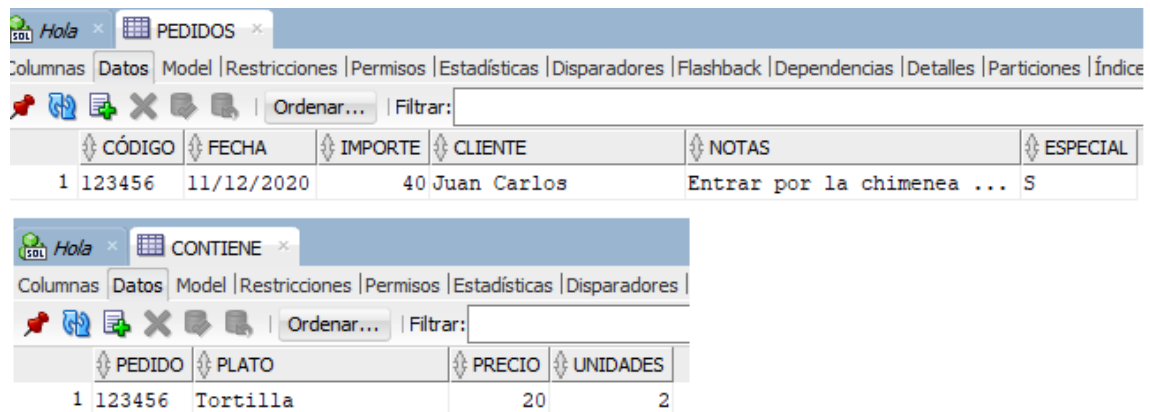
INSERT INTO pedidos VALUES...

The screenshot shows a database management tool with two tabs: PEDIDOS and CONTIENE. The PEDIDOS tab is active, displaying a table with columns: CÓDIGO, FECHA, IMPORTE, CLIENTE, NOTAS, and ESPECIAL. A single row is visible with the following data: 1, 123456, 11/12/2020, 0, Juan Carlos, Entrar por la chimenea ..., S. The CONTIENE tab is also visible, showing a table with columns: PEDIDO, PLATO, PRECIO, and UNIDADES.

CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1 123456	11/12/2020	0	Juan Carlos	Entrar por la chimenea ...	S

PEDIDO	PLATO	PRECIO	UNIDADES
--------	-------	--------	----------

INSERT INTO contiene VALUES...

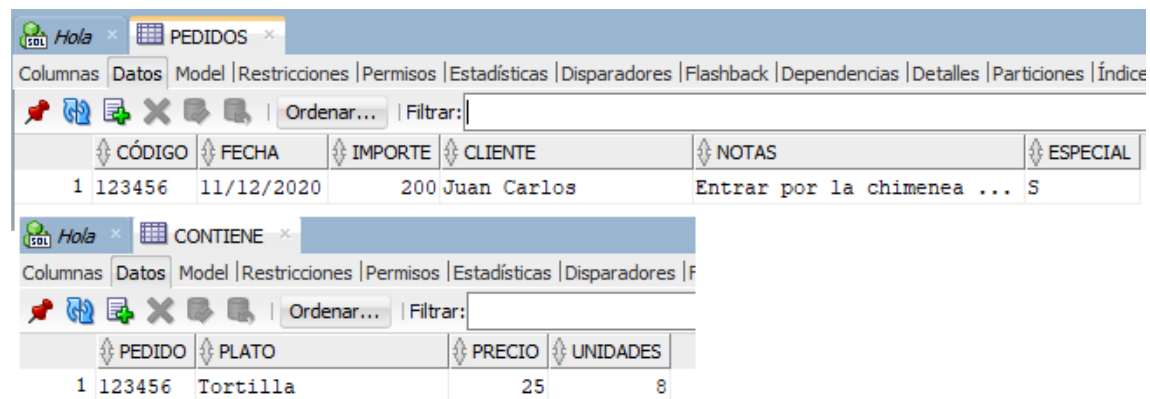


The screenshot shows two SQL Server Enterprise Manager windows. The top window, titled 'Hola' and 'PEDIDOS', displays a table with columns: CÓDIGO, FECHA, IMPORTE, CLIENTE, NOTAS, and ESPECIAL. It contains one record with values: 1, 123456, 11/12/2020, 40 Juan Carlos, Entrar por la chimenea ..., and S. The bottom window, also titled 'Hola' and 'CONTIENE', displays a table with columns: PEDIDO, PLATO, PRECIO, and UNIDADES. It contains one record with values: 1, 123456, Tortilla, 20, and 2.

	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1	123456	11/12/2020	40	Juan Carlos	Entrar por la chimenea ...	S

	PEDIDO	PLATO	PRECIO	UNIDADES
1	123456	Tortilla	20	2

UPDATE contiene SET...

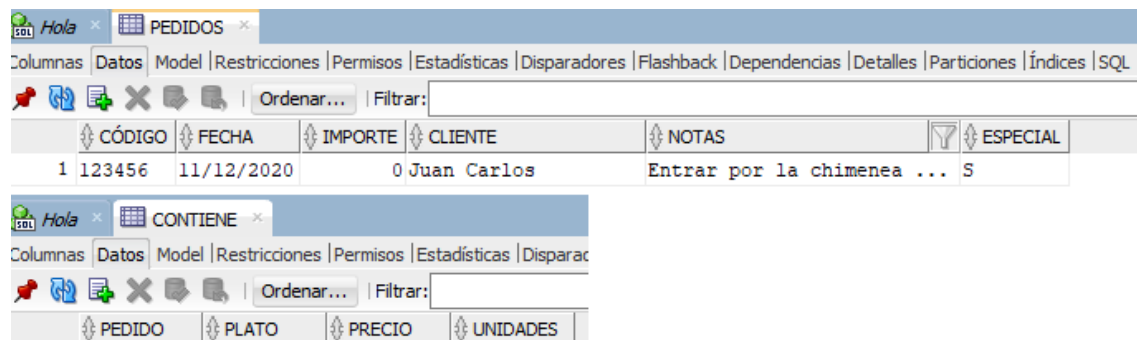


The screenshot shows the same two SQL Server Enterprise Manager windows as before. In the 'PEDIDOS' table, the 'IMPORTE' value has been updated from 40 to 200. In the 'CONTIENE' table, the 'PRECIO' value has been updated from 20 to 25 and the 'UNIDADES' value has been updated from 2 to 8.

	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1	123456	11/12/2020	200	Juan Carlos	Entrar por la chimenea ...	S

	PEDIDO	PLATO	PRECIO	UNIDADES
1	123456	Tortilla	25	8

DELETE FROM contiene...

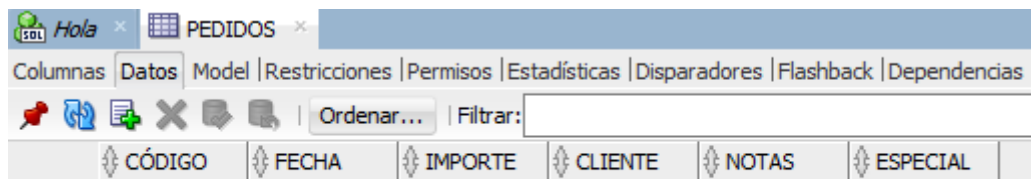


The screenshot shows the same two SQL Server Enterprise Manager windows. In the 'PEDIDOS' table, the 'IMPORTE' value has been updated from 40 to 0. In the 'CONTIENE' table, the 'PRECIO' value has been updated from 25 to 0 and the 'UNIDADES' value has been updated from 8 to 0.

	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1	123456	11/12/2020	0	Juan Carlos	Entrar por la chimenea ...	S

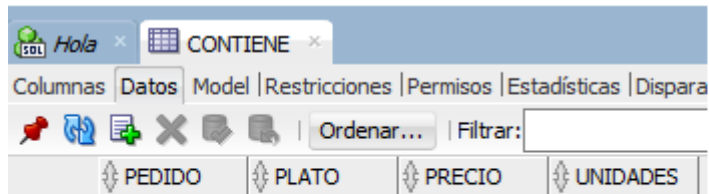
	PEDIDO	PLATO	PRECIO	UNIDADES
1	123456	Tortilla	0	0

DELETE FROM pedidos



SQL Developer interface showing the 'PEDIDOS' table structure. The table has columns: CÓDIGO, FECHA, IMPORTE, CLIENTE, NOTAS, and ESPECIAL.

CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
--------	-------	---------	---------	-------	----------



SQL Developer interface showing the 'CONTIENE' table structure. The table has columns: PEDIDO, PLATO, PRECIO, and UNIDADES.

PEDIDO	PLATO	PRECIO	UNIDADES
--------	-------	--------	----------

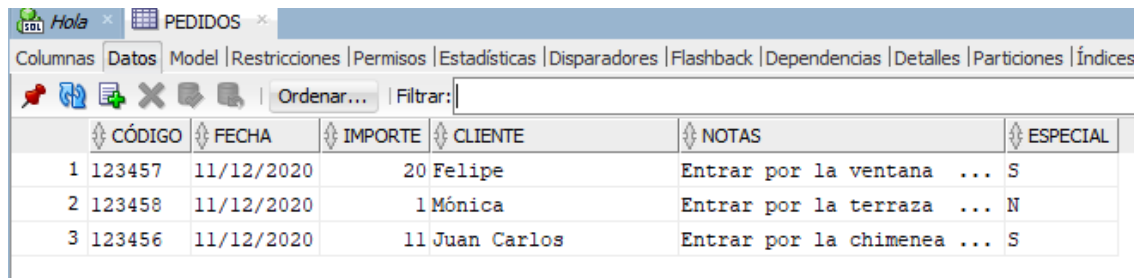
3

```
SOLUCIÓN 1 (FOR EACH ROW):
ERROR: ORA-04091: la tabla JULLAMAS.PEDIDOS está mutando,
puede que el disparador/la función no puedan verla
La tabla está mutando porque estamos intentando modificarla con un disparador
cuando realizamos un UPDATE sobre ella. Para solucionarlo vamos a usar INSTEAD OF
CREATE OR REPLACE TRIGGER tr_especial
AFTER INSERT OR UPDATE ON pedidos
FOR EACH ROW
DECLARE
    v_media pedidos.importe%TYPE;
BEGIN
    SELECT AVG(importe) INTO v_media FROM pedidos;
    UPDATE pedidos SET especial = CASE WHEN importe > v_media THEN 'S' ELSE 'N' END;
END tr_especial;
*/

--SOLUCIÓN 2 (INSTEAD OF):
CREATE OR REPLACE TRIGGER tr_especial
INSTEAD OF INSERT OR UPDATE ON vista_pedidos
FOR EACH ROW
DECLARE
    v_media pedidos.importe%TYPE;
    v_num NATURAL;
    v_sum pedidos.importe%TYPE;
BEGIN
    SELECT COUNT(importe), SUM(importe) INTO v_num, v_sum FROM pedidos;
    IF INSERTING THEN
        v_media := (v_sum + :NEW.importe) / (v_num + 1);
    ELSIF UPDATING THEN
        v_media := (v_sum - :OLD.importe + :NEW.importe) / (v_num);
        DELETE FROM pedidos WHERE código = :OLD.código;
    END IF;
    INSERT INTO pedidos VALUES (:NEW.código, :NEW.fecha, :NEW.importe,
    :NEW.cliente, :NEW.notas, :NEW.especial);
    UPDATE pedidos SET especial = CASE WHEN importe > v_media THEN 'S' ELSE 'N' END;
END tr_especial;
/

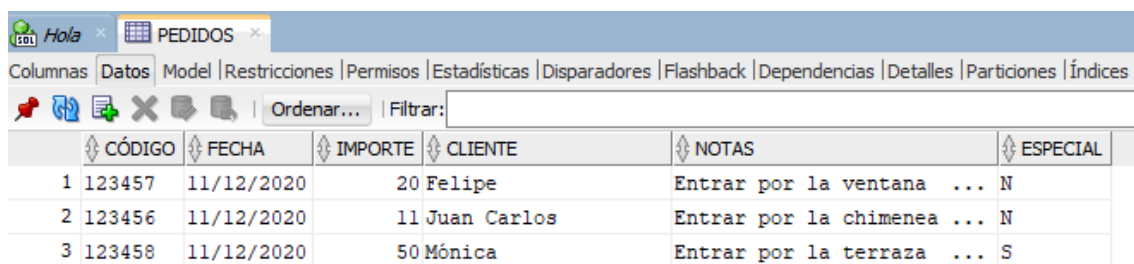
BEGIN
    INSERT INTO vista_pedidos VALUES ('123456', '11/12/2020', 11, 'Juan Carlos', 'Entrar por la chimenea', 'N');
    INSERT INTO vista_pedidos VALUES ('123457', '11/12/2020', 20, 'Felipe', 'Entrar por la ventana', 'N');
    INSERT INTO vista_pedidos VALUES ('123458', '11/12/2020', 1, 'Mónica', 'Entrar por la terraza', 'N');
    UPDATE vista_pedidos SET importe = 50 WHERE código = '123458';
END;
```

(tras las tres primeras inserciones)



	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1	123457	11/12/2020	20	Felipe	Entrar por la ventana ...	S
2	123458	11/12/2020	1	Mónica	Entrar por la terraza ...	N
3	123456	11/12/2020	11	Juan Carlos	Entrar por la chimenea ...	S

(tras la actualización)



	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS	ESPECIAL
1	123457	11/12/2020	20	Felipe	Entrar por la ventana ...	N
2	123456	11/12/2020	11	Juan Carlos	Entrar por la chimenea ...	N
3	123458	11/12/2020	50	Mónica	Entrar por la terraza ...	S

Texto plano

DROP TABLE contiene;

DROP TABLE pedidos;

DROP TABLE auditoría;

DROP VIEW vista_pedidos;

CREATE TABLE pedidos(código CHAR(6) NOT NULL, fecha CHAR(10) NOT NULL,

importe NUMBER(6,2) DEFAULT 0, cliente CHAR(20) NOT NULL, notas CHAR(1024),

especial CHAR(1) NOT NULL, CHECK (especial in('S','N')), CHECK (importe>=0),

PRIMARY KEY(código));

CREATE VIEW vista_pedidos AS SELECT * FROM pedidos;

CREATE TABLE contiene(pedido CHAR(6) REFERENCES pedidos(código), plato

```
CHAR(20) NOT NULL, precio NUMBER(6,2) NOT NULL, unidades  
NUMBER(2,0) NOT NULL,  
CHECK (precio>=0), CHECK (unidades>0), PRIMARY KEY(pedido,plato));
```

```
CREATE TABLE auditoría(operación CHAR(6) NOT NULL, tabla CHAR(50)  
NOT NULL,  
fecha CHAR(10) NOT NULL, hora CHAR(8) NOT NULL,
```

```
CHECK (operación in('DELETE','INSERT','UPDATE')));  
--Esta tabla no contiene Primary key porque puedes hacer operaciones  
--simultaneas del mismo tipo en una misma tabla y en un mismo instante
```

```
CREATE OR REPLACE TRIGGER tr_pedidos  
AFTER INSERT OR DELETE OR UPDATE ON pedidos  
BEGIN  
    IF DELETING THEN  
        INSERT INTO auditoría VALUES('DELETE','pedidos',  
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));  
    ELSIF INSERTING THEN  
        INSERT INTO auditoría VALUES('INSERT','pedidos',  
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));  
    ELSIF UPDATING THEN  
        INSERT INTO auditoría VALUES('UPDATE','pedidos',  
            to_char(sysdate,'dd/mm/yyyy'),to_char(sysdate,'hh:mi:ss'));  
    END IF;  
END tr_pedidos;  
/
```

```
BEGIN  
    INSERT INTO pedidos VALUES('123456','11/12/2020',0,'Juan Carlos',  
        'Entrar por la chimenea','S');  
    UPDATE pedidos SET importe=importe+1 WHERE código='123456';  
    DELETE FROM pedidos WHERE código='123456';  
END;  
/
```

```
CREATE OR REPLACE TRIGGER tr_contiene
```

```

AFTER INSERT OR DELETE OR UPDATE ON contiene
FOR EACH ROW
BEGIN
    IF DELETING THEN
        UPDATE pedidos SET importe= importe - :OLD.precio*:OLD.unidades
        WHERE :OLD.pedido=código;
    ELSIF INSERTING THEN
        UPDATE pedidos SET importe=:NEW.precio*:NEW.unidades
        WHERE :NEW.pedido=código;
    ELSIF UPDATING THEN
        UPDATE pedidos SET importe=importe-:OLD.precio*:OLD.unidades +
        :NEW.precio*:NEW.unidades WHERE :OLD.pedido=código;
    END IF;
END tr_contiene;
/
BEGIN
    INSERT INTO pedidos VALUES('123456','11/12/2020',0,'Juan Carlos',
    'Entrar por la chimenea','S');
    INSERT INTO contiene VALUES('123456','Tortilla',20,2);
    UPDATE contiene SET precio=precio+5, unidades = unidades + 6
    WHERE pedido='123456'and plato='Tortilla';
    DELETE FROM contiene WHERE pedido='123456';
    DELETE FROM pedidos WHERE código='123456';
END;
/

```

/*

SOLUCIÓN 1 (FOR EACH ROW):

ERROR: ORA-04091: la tabla JULLAMAS.PEDIDOS está mutando,
puede que el disparador/la función no puedan verla

La tabla está mutando porque estamos intentando modificarla con un
disparador

cuando realizamos un UPDATE sobre ella. Para solucionarlo vamos a usar
INSTEAD OF

CREATE OR REPLACE TRIGGER tr_especial

AFTER INSERT OR UPDATE ON pedidos

FOR EACH ROW

DECLARE

v_media pedidos.importe%TYPE;


```

BEGIN
    SELECT AVG(importe) INTO v_media FROM pedidos;
    UPDATE pedidos SET especial = CASE WHEN importe>v_media THEN 'S'
ELSE 'N' END;
END tr_especial;
*/
--SOLUCIÓN 2 (INSTEAD OF):
CREATE OR REPLACE TRIGGER tr_especial
INSTEAD OF INSERT OR UPDATE ON vista_pedidos
FOR EACH ROW
DECLARE
    v_media pedidos.importe%TYPE;
    v_num NATURAL;
    v_sum pedidos.importe%TYPE;
BEGIN
    SELECT COUNT(importe),SUM(importe) INTO v_num,v_sum FROM
pedidos;
    IF INSERTING THEN
        v_media:=(v_sum+:NEW.importe)/(v_num+1);
    ELSIF UPDATING THEN
        v_media:=(v_sum-:OLD.importe+:NEW.importe)/(v_num);
        DELETE FROM pedidos WHERE código=:OLD.código;
    END IF;
    INSERT INTO pedidos VALUES(:NEW.código,:NEW.fecha,:NEW.importe,
:NEW.cliente,:NEW.notas,:NEW.especial);
    UPDATE pedidos SET especial = CASE WHEN importe>v_media THEN 'S'
ELSE 'N' END;
END tr_especial;
/
BEGIN
    INSERT INTO vista_pedidos VALUES('123456','11/12/2020',11,'Juan
Carlos','Entrar por la chimenea','N');
    INSERT INTO vista_pedidos
VALUES('123457','11/12/2020',20,'Felipe','Entrar por la ventana','N');
    INSERT INTO vista_pedidos
VALUES('123458','11/12/2020',1,'Mónica','Entrar por la terraza','N');
    UPDATE vista_pedidos SET importe=50 WHERE código='123458';
END;
/

```