

-1

```
CREATE TABLE pedidos (  
    código CHAR(6) PRIMARY KEY,  
    fecha CHAR(10) NOT NULL,  
    importe NUMBER(6,2),  
    cliente CHAR(20) ,  
    notas CHAR(1024)  
);
```

Table PEDIDOS creado.

```
CREATE TABLE contiene (  
    pedido CHAR(6) ,  
    plato CHAR(10) ,  
    precio NUMBER(6,2),  
    unidades NUMBER(2,0),  
    PRIMARY KEY (pedido, plato)  
);
```

Table CONTIENE creado.

```
CREATE TABLE auditoría (  
    operación CHAR(6),  
    tabla CHAR(50),  
    fecha CHAR(10),  
    hora CHAR(8)  
);
```

/

Table AUDITORÍA creado.

```
CREATE OR REPLACE TRIGGER trigger_pedidos  
AFTER INSERT OR UPDATE OR DELETE 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 trigger_pedidos;
```

```
/
```

```
Trigger TRIGGER_PEDIDOS compilado
```

```
INSERT INTO pedidos VALUES('123456','20/11/1929',10.0,'Julio del Valle','Grandioso');
```

```
/
```

```
1 fila insertadas.
```

```
UPDATE pedidos SET importe=importe + 1;
```

```
/
```

```
1 fila actualizadas.
```

```
DELETE FROM pedidos;
```

```
/
```

```
1 fila eliminado
```

```
-2
```

```
CREATE OR REPLACE TRIGGER trigger_contiene
```

```
AFTER INSERT OR UPDATE OR DELETE ON contiene
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF DELETING THEN
```

```
UPDATE pedidos
```

```
SET importe = importe - (:OLD.precio);
```

```
ELSIF INSERTING THEN
```

```
UPDATE pedidos
```

```
SET importe = importe + (:NEW.precio) * (:NEW.unidades));
```

ELSIF UPDATING THEN

UPDATE pedidos

SET importe = importe - (:OLD.precio) + (:NEW.precio);

END IF;

END trigger_contiene;

/

Trigger TRIGGER_CONTIENE compilado

-3

- a

SET TIMING ON

/

CREATE INDEX index_pedidos

ON pedidos (cliente);

/

Index INDEX_PEDIDOS creado.

Transcurrido: 00:00:00.014

- b

BEGIN

FOR i in 0..300000 LOOP

INSERT INTO pedidos VALUES(i,'06/01/2015',10.0,'C' || i, '');

END LOOP;

END;

/

Procedimiento PL/SQL terminado correctamente.

Transcurrido: 00:01:44.763

- c

select * from pedidos where cliente = 'C300000';

DROP INDEX index_pedidos;

select * from pedidos where cliente = 'C300000';

/

>>Query Run In:Resultado de la Consulta 0

Transcurrido: 00:00:00.341

Index INDEX_PEDIDOS borrado.

Transcurrido: 00:00:00.018

>>Query Run In:Resultado de la Consulta 1

Transcurrido: 00:00:00.346

- d

select * from pedidos where código ='300000';

ALTER TABLE pedidos DROP PRIMARY KEY;

select * from pedidos where código ='300000';

/

Transcurrido: 00:00:00.039

>>Query Run In:Resultado de la Consulta 3

Transcurrido: 00:00:00.336

- e

create view pedidosuno as

select *

from pedidos

where código ='300000';

/

View PEDIDOSUNO creado.

Transcurrido: 00:00:00.012

create materialized view pedidosdos as

select *

from pedidos

where código ='300000';

/

Materialized view PEDIDOSDOS creado.

Transcurrido: 00:00:12.372

CREATE INDEX index_pedidosuno

ON pedidosuno (cliente);

/

Informe de error -

ORA-01702: una vista no es apropiada aquí

**Cause: Among other possible causes, this message will be produced if an attempt was made to define an Editioning View over a view.*

**Action: An Editioning View may only be created over a base table.*

Transcurrido: 00:00:00.008

CREATE INDEX index_pedidosdos

ON pedidosdos (cliente);

/

Index INDEX_PEDIDOSDOS creado.

Transcurrido: 00:00:00.026