```
SELECT TABLE_NAME FROM USER_TABLES;

DROP TABLE auditoría CASCADE CONSTRAINTS;
DROP TABLE contiene CASCADE CONSTRAINTS;
DROP TABLE pedidos CASCADE CONSTRAINTS;

CREATE TABLE pedidos(
código CHAR(6),
fecha CHAR(10),
importe NUMBER(6,2),
cliente CHAR(20),
notas CHAR(1024),
PRIMARY KEY(código)
);

--Table PEDIDOS creado.

CREATE TABLE contiene(
pedido CHAR(6),
plato CHAR(20),
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.

--Apartado 1

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

--Trigger TRIGGER_PEDIDOS compilado

--Insertar
INSERT INTO pedidos(código, fecha, importe, cliente, notas) VALUES ('3', '2018', 6, 'C1',
'Prueba');

--1 fila insertadas.

--Actualizar
```

```
UPDATE pedidos
SET fecha = '2019'
WHERE código = '3';

--1 fila actualizadas.

--Borrar
DELETE FROM pedidos
WHERE código = '3';

--1 fila eliminado

SELECT operación, tabla, fecha, hora FROM auditoría;

/*
INSERT  pedidos                         29/11/2018      12:31:09
UPDATE          pedidos                         29/11/2018      12:32:58
DELETE pedidos                          29/11/2018      12:33:24*/

--Apartado 2

CREATE OR REPLACE TRIGGER trigger_contiene
AFTER INSERT OR DELETE OR UPDATE ON contiene
FOR EACH ROW
BEGIN
IF INSERTING THEN
UPDATE pedidos
SET pedidos.importe = pedidos.importe + (:NEW.precio * :NEW.unidades)
WHERE pedidos.código = :NEW.pedido;
ELSIF DELETING THEN
UPDATE pedidos
```

```
SET pedidos.importe = pedidos.importe - (:OLD.precio * :OLD.unidades)

WHERE pedidos.código = :OLD.pedido;

ELSIF UPDATING THEN

UPDATE pedidos

SET pedidos.importe = pedidos.importe - (:OLD.precio * :OLD.unidades) + (:NEW.precio *
:NEW.unidades)

WHERE pedidos.código = :NEW.pedido;

END IF;

END;


--Trigger TRIGGER_CONTIENE compilado


--Poner datos para comprobarlo

--Insertar

INSERT INTO pedidos(código, fecha, importe, cliente, notas) VALUES ('3', '2018', 6, 'C1',
'Prueba');

INSERT INTO contiene(pedido, plato, precio, unidades) VALUES ('3', '123456789PWD', 6, 2);


--1 fila insertadas.

--1 fila insertadas.


SELECT importe FROM pedidos WHERE código = '3';


--18


--Actualizar

UPDATE contiene

SET precio = 7

WHERE pedido = '3';


--1 fila actualizadas.
```

```
SELECT importe FROM pedidos WHERE código = '3';


--20


--Borrar
DELETE FROM contiene
WHERE pedido = '3';


--1 fila eliminado


SELECT importe FROM pedidos WHERE código = '3';


--6


--Apartado 3


--a)
CREATE INDEX index_pedidos ON pedidos(cliente);


--Index INDEX_PEDIDOS creado.


--b)
CREATE OR REPLACE PROCEDURE rellenarTablaPedidos IS inicio NUMBER:= 1; fin INTEGER:=
50000; v_cliente pedidos.cliente%TYPE; BEGIN FOR I IN inicio..fin LOOP v_cliente:= 'C' || I;
INSERT INTO pedidos VALUES (I,'06/01/2015',10.0,v_cliente,' '); END LOOP; END;
--ALTER DATABASE 'D:\oracle\EMPRESACC05' AUTOEXTEND ON MAXSIZE 500M;


CREATE OR REPLACE PROCEDURE rellenarTablaPedidos IS
   inicio NUMBER:= 1;
   fin INTEGER:= 50000;
   v_cliente pedidos.cliente%TYPE;
```

```
BEGIN
FOR I IN inicio..fin LOOP
v_cliente:= 'C' || I;
INSERT INTO pedidos VALUES (I,'06/01/2015',10.0,v_cliente,' ');
END LOOP;
END;

--Procedure RELLENARTABLAPEDIDOS compilado

EXECUTE rellenarTablaPedidos;

--Procedimiento PL/SQL terminado correctamente.

--c)

SET TIMING ON;

SELECT codigo, fecha, importe, cliente, notas FROM pedidos WHERE pedidos.cliente =
'C50000';

-- 00:00:00.04

DROP INDEX index_pedidos;
/*
Index INDEX_PEDIDOS borrado.

00:00:00.06*/

SELECT codigo, fecha, importe, cliente, notas FROM pedidos WHERE pedidos.cliente =
'C50000';

--00:00:00.07
```

--d)

```
CREATE VIEW vista WITH SCHEMABINDING AS SELECT * FROM pedidos;
CREATE INDEX vistaIndice ON vista(cliente); --No se puede crear

CREATE MATERIALIZED VIEW vistaMaterializada(codigo, fecha, importe, cliente, notas)
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND AS (SELECT * FROM pedidos);
--Materialized view VISTAMATERIALIZADA creado.

CREATE INDEX vistaMaterializadaIndice ON vistaMaterializada(cliente);
--Index VISTAMATERIALIZADAINDICE creado.
```