



## TEMA 6. DISEÑO FÍSICO. DQL EVALUABLE4

BASES DE DAATOS  
CFGs DAW

Autor: Paco Aldarias Raya

[paco.aldarias@gmail.com](mailto:paco.aldarias@gmail.com)

2019/2020

26/03/20

Versión:200326.2022

## Licencia



**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

## Revisión

25/3/2020. Añadidas más soluciones a todos los problemas.

## ÍNDICE DE CONTENIDO

<b>1.CONSULTA 1.....</b>	<b>12</b>
<b>2.CONSULTA 2.....</b>	<b>13</b>
<b>3.CONSULTA 3.....</b>	<b>13</b>
<b>4.CONSULTA 4.....</b>	<b>14</b>
<b>5.CONSULTA 5.....</b>	<b>15</b>
<b>6.CONSULTA 6.....</b>	<b>15</b>

## UD06. EVALUABLE4

La siguiente actividad evaluable, permite repasar el tema 6 de SQL. De deberá entregar un fichero llamado nombre\_alumnoevalu4.sql que contenga las soluciones en sql de todas las consultas. Ejemplo del fichero nombre\_alumnoevalu4solucion.sql

- Consulta 1.

```
select * ....
```

- Consulta 2.

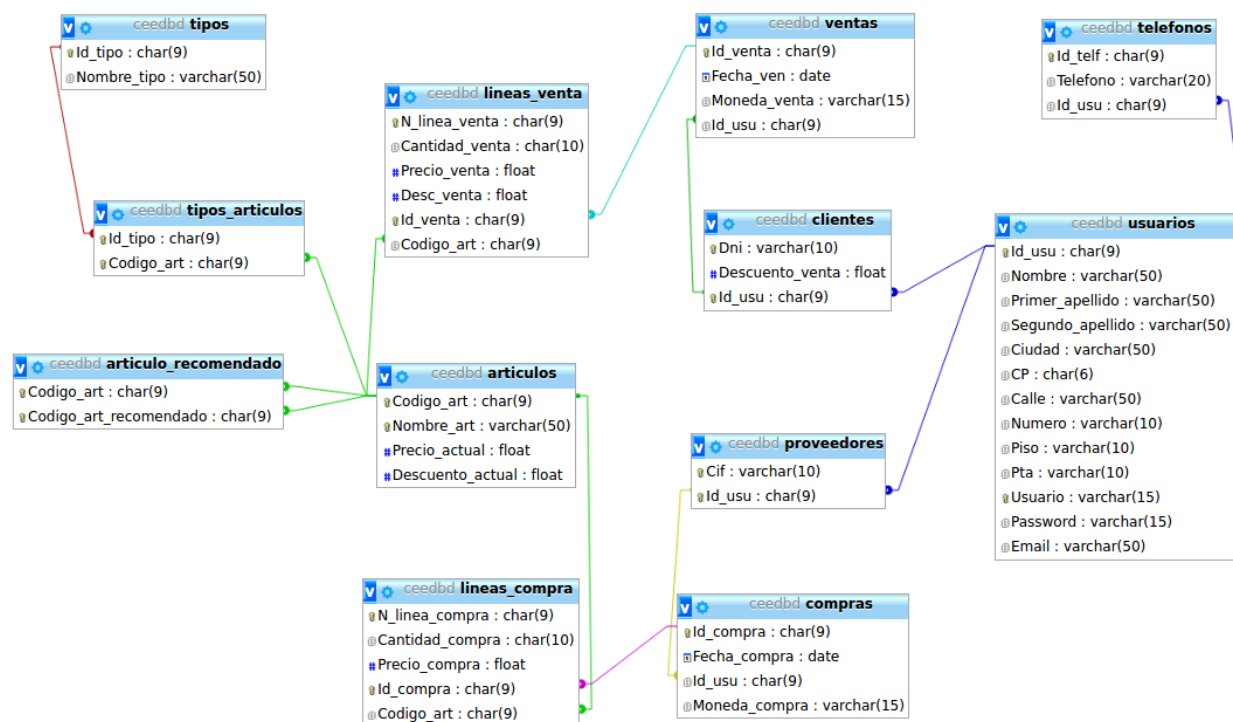
```
select * ....
```

Las soluciones podrán ser: select, select con update, create table as ..., create view .... Es decir todo lo visto en el tema.

Calificación:

- Preguntas: 1,5 valen 1 punto.
- Preguntas: 2,3, 4, 6 valen 2 puntos.

El diagrama del diseño físico sería:



*Imagen 1:*

Las tablas y los datos seria:

```
DROP TABLE IF EXISTS articulo_recomendado;
```

```
DROP TABLE IF EXISTS lineas_compra;
```

```
DROP TABLE IF EXISTS lineas_venta;
```

```
DROP TABLE IF EXISTS telefonos;
```

```
DROP TABLE IF EXISTS tipos_articulos;
```

```
DROP TABLE IF EXISTS tipos;
```

```
DROP TABLE IF EXISTS articulos;
```

```
DROP TABLE IF EXISTS ventas;
```

```
DROP TABLE IF EXISTS compras;
```

```
DROP TABLE IF EXISTS clientes;
```

```
DROP TABLE IF EXISTS proveedores;
```

```
DROP TABLE IF EXISTS usuarios;
```

```
--
```

```
CREATE TABLE usuarios(
```

```
Id_usu CHAR(9),
```

```
Nombre VARCHAR(50),
```

```
Primer_apellido VARCHAR(50),
```

```
Segundo_apellido VARCHAR(50),
```

```
Ciudad VARCHAR(50),
```

```
CP CHAR(6),
```

```
Calle VARCHAR(50),
```

```
Numero VARCHAR(10),
```

```
Piso VARCHAR(10),
```

```
Pta VARCHAR(10),
```

```
Usuario VARCHAR(15) NOT NULL UNIQUE,
```

```
Password VARCHAR(15) NOT NULL,
```

```
Email VARCHAR(50),
```

```
CONSTRAINT USU_IDU_PK PRIMARY KEY(Id_usu)
```

```
);
```

```
--
```

```
INSERT INTO usuarios (Id_usu, Nombre, Primer_apellido, Segundo_apellido,
```

```
Ciudad, CP, Calle, Numero, Piso, Pta, Usuario, Password, Email)
```

```
VALUES ('001', 'José', 'Alfaro', 'García', 'Albacete', '02006',
'María Marín', '57', '03', 'B', 'Pepin33', 'pepin', 'pepin@hotmail.es'),
('002', 'Miguel', 'Perez', 'Pena', 'Albacete', '02007',
'Perez Pastor', '41', '02', 'A', 'Mike47', '1234', 'mikel@hotmail.es'),
('003', 'Ana', 'Ponce', 'García', 'Madrid', '28001',
'Gran Via', '07', '10', 'F', 'Anika12', '0123456', 'anika@hotmail.es'),
('004', 'Antonio', 'Fernandez', 'Ruiz', 'Valencia', '46008',
'San Vicente', '11', '5', 'F', 'antru', '7777777', 'antru@gmail.es')
;
--

CREATE TABLE telefonos(
Id_telf CHAR(9),
Telefono VARCHAR(20),
Id_usu CHAR(9),
CONSTRAINT TEL_IDT_PK PRIMARY KEY(Id_telf),
CONSTRAINT TEL_IDU_FK FOREIGN KEY(Id_usu) REFERENCES usuarios (Id_usu)
ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO telefonos (Id_telf, Telefono, Id_usu)
VALUES ('101', '967334546', '001'),
('102', '967348632', '002'),
('103', '914562532', '003');
```

```
CREATE TABLE clientes(
Dni VARCHAR(10) NOT NULL UNIQUE,
Descuento_venta FLOAT,
Id_usu CHAR(9),
CONSTRAINT CLI_IDU_PK PRIMARY KEY(Id_usu),
CONSTRAINT CLI_IDU_FK FOREIGN KEY(Id_usu) REFERENCES usuarios (Id_usu)
ON DELETE CASCADE ON UPDATE CASCADE
```

);

```
INSERT INTO clientes (Dni, Descuento_venta, Id_usu)
VALUES ('48065241C', 0, '001'),
('20154478C', 0, '002'),
('68941252B', 0, '003'),
('12345678D', 0, '004');
```

```
CREATE TABLE articulos(
Codigo_art CHAR(9),
Nombre_art VARCHAR(50) NOT NULL UNIQUE,
Precio_actual FLOAT,
Descuento_actual FLOAT,
CONSTRAINT ART_COA_PK PRIMARY KEY(Codigo_art)
);
```

```
INSERT INTO articulos (Codigo_art, Nombre_art, Precio_actual, Descuento_actual)
VALUES ('1001', 'Muñeca', 20.5, 5),
('1002', 'Coche de carreras', 15.3, 2),
('1003', 'Tangram', 30.5, 1),
('1004', 'Lampara mesa', 60.65, 0),
('1005', 'Cuadro con flores', 80.5, 0),
('1006', 'Centro de mesa', 80.5, 0);
```

```
CREATE TABLE ventas(
Id_venta CHAR(9),
Fecha_ven DATE NOT NULL,
Moneda_venta VARCHAR(15) NOT NULL,
Id_usu CHAR(9) NOT NULL,
CONSTRAINT VEN_IDV_PK PRIMARY KEY(Id_venta),
CONSTRAINT VEN_IDU_FK FOREIGN KEY(Id_usu) REFERENCES clientes (Id_usu)
ON DELETE NO ACTION ON UPDATE NO ACTION
```

);

```
INSERT INTO ventas (Id_venta, Fecha_ven, Moneda_venta, Id_usu)
VALUES ('2002', '2019-12-15', 'EUROS', '003'),
('2001', '2019-12-14', 'EUROS', '001'),
('2003', '2019-12-13', 'DOLARES', '002');
```

```
CREATE TABLE lineas_venta(
N_linea_venta CHAR(9),
Cantidad_venta CHAR(10) NOT NULL,
Precio_venta FLOAT NOT NULL,
Desc_venta FLOAT,
Id_venta CHAR(9),
Codigo_art CHAR(9) NOT NULL,
CONSTRAINT LIV_NLV_PK PRIMARY KEY(N_linea_venta, Id_venta),
CONSTRAINT LIV_IDV_FK FOREIGN KEY(Id_venta) REFERENCES ventas (Id_venta)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT LIV_COA_FK FOREIGN KEY(Codigo_art) REFERENCES articulos (Codigo_art)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO lineas_venta (N_linea_venta, Cantidad_venta, Precio_venta, Desc_venta, Id_venta,
Codigo_art)
VALUES ('001', '2', 20.5, 0, '2001', '1001'),
('002', '2', 15.3, 0, '2001', '1002'),
('003', '1', 30.5, 0, '2001', '1003');
```

```
CREATE TABLE proveedores(
Cif VARCHAR(10) NOT NULL UNIQUE,
Id_usu CHAR(9),
CONSTRAINT PRO_IDU_PK PRIMARY KEY(Id_usu),
```



```
CONSTRAINT PRO_IDU_FK FOREIGN KEY(Id_usu) REFERENCES usuarios(Id_usu)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO proveedores (Cif, Id_usu)
VALUES ('B01234567C', '001'),
('B23541253N', '002'),
('B89546325S', '003');
```

```
CREATE TABLE compras(
Id_compra CHAR(9),
Fecha_compra DATE NOT NULL,
Id_usu CHAR(9) NOT NULL,
Moneda_compra VARCHAR(15) NOT NULL,
CONSTRAINT COM_IDC_PK PRIMARY KEY(Id_compra),
CONSTRAINT COM_IDU_FK FOREIGN KEY(Id_usu) REFERENCES proveedores(Id_usu)
ON DELETE NO ACTION ON UPDATE NO ACTION
);
```

```
INSERT INTO compras (Id_compra, Fecha_compra, Moneda_compra, Id_usu)
VALUES ('3001', '2019-12-10', 'EUROS', '001'),
('3002', '2019-12-11', 'EUROS', '002'),
('3003', '2019-12-12', 'DOLARES', '003');
```

```
CREATE TABLE lineas_compra(
N_linea_compra CHAR(9),
Cantidad_compra CHAR(10) NOT NULL,
Precio_compra FLOAT NOT NULL,
Id_compra CHAR(9),
Codigo_art CHAR(9) NOT NULL,
```

```
CONSTRAINT LIC_NLC_PK PRIMARY KEY(N_linea_compra, Id_compra),
CONSTRAINT LIC_IDC_FK FOREIGN KEY(Id_compra) REFERENCES compras (Id_compra)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT LIC_COA_FK FOREIGN KEY(Codigo_art) REFERENCES articulos (Codigo_art)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO lineas_compra (N_linea_compra, Cantidad_compra, Precio_compra,
Id_compra, Codigo_art)
VALUES ('001', '2', '10.5', '3001', '1001'),
('002', '4', '10.2', '3001', '1002'),
('003', '1', '10.5', '3001', '1003'),
('001', '1', '2', '3002', '1001');
```

```
CREATE TABLE tipos(
Id_tipo CHAR(9),
Nombre_tipo VARCHAR(50) NOT NULL,
CONSTRAINT TIP_IDT_PK PRIMARY KEY(Id_tipo)
);
```

```
INSERT INTO tipos (Id_tipo, Nombre_tipo)
VALUES ('4001', 'preescolar'),
('4002', 'movimiento'),
('4003', 'educativos'),
('4004', 'decoración');
```

```
CREATE TABLE tipos_articulos(
Id_tipo CHAR(9),
Codigo_art CHAR(9),
```

```
CONSTRAINT TIA_IDC_PK PRIMARY KEY (Id_tipo,Codigo_art),
CONSTRAINT TIA_IDT_FK FOREIGN KEY (Id_tipo) REFERENCES tipos(Id_tipo)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT TIA_COA_FK FOREIGN KEY (Codigo_art) REFERENCES articulos(Codigo_art)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO tipos_articulos (Id_tipo, Codigo_art)
VALUES ('4001', '1001'),
('4002', '1002'),
('4003', '1003'),
('4004', '1004'),
('4004', '1005'),
('4004', '1006');
```

```
CREATE TABLE articulo_recomendado(
Codigo_art CHAR(9),
Codigo_art_recomendado CHAR(9),
CONSTRAINT ARR_CAR_PK PRIMARY KEY (Codigo_art, Codigo_art_recomendado),
CONSTRAINT ARR_COA_FK FOREIGN KEY (Codigo_art) REFERENCES articulos(Codigo_art),
CONSTRAINT ARR_COAF_FK FOREIGN KEY (Codigo_art_recomendado) REFERENCES
articulos(Codigo_art)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO articulo_recomendado (Codigo_art, Codigo_art_recomendado)
VALUES ('1004', '1005'),
('1004', '1006'),
('1001', '1002');
```

## 1. CONSULTA 1

Obtener el id de la compra, el id del usuario, el nombre del usuario y el importe de cada compra, ordenado por el id de la compra ascendente. Como se muestra en la imagen:

+ Opciones			
id_compra	id_usu	nombre	importe_total
3001	001	José	61.79999923706055
3002	002	Miguel	10.5

Solución 1:

```
select com.id_compra, com.id_usu, usu.nombre, sum(lin.cantidad_compra * lin.precio_compra)
as importe_total
from compras as com, lineas_compra as lin, usuarios as usu
where com.id_compra = lin.id_compra and com.id_usu=usu.id_usu
group by com.id_compra
order by com.id_compra;
```

Solución 2:

```
SELECT
  COMPRAS.id_compra,
  USUARIOS.id_usu,
  USUARIOS.nombre,
  sum(LINEAS_COMPRA.precio_compra * CAST(cantidad_compra AS int)) AS importe_total
FROM COMPRAS
  INNER JOIN USUARIOS ON USUARIOS.id_usu = COMPRAS.id_usu
  INNER JOIN LINEAS_COMPRA ON LINEAS_COMPRA.id_compra = COMPRAS.id_compra
GROUP BY COMPRAS.id_compra, USUARIOS.id_usu
ORDER BY COMPRAS.id_compra ASC;
```

Solución 3:










```
select distinct c.id_compra, u.id_usu, u.nombre, sum(cast(l.cantidad_compra as int4) *
l.precio_compra) as importe_total
from usuarios u natural join compras c natural join lineas_compra l
group by c.id_compra, u.id_usu
order by id_compra;
```

## 2. CONSULTA 2

-- Mostrar del articulo el código, nombre y el importe total comprado en todas las compras en

cualquier fecha. Es decir, debe mostrar el total de compras de cada artículo. Como se muestra en la imagen:

+ Opciones

<div>←T→</div>	Codigo_art ▲ 1	Nombre_art	importe_comprado
<div><div><input type="checkbox"/></div><div><div> Editar</div><div> Copiar</div><div> Borrar</div></div></div> <td>1001</td> <td>Muñeca</td> <td>23</td>	1001	Muñeca	23
<div><div><input type="checkbox"/></div><div><div> Editar</div><div> Copiar</div><div> Borrar</div></div></div> <td>1002</td> <td>Coche de carreras</td> <td>40.79999923706055</td>	1002	Coche de carreras	40.79999923706055
<div><div><input type="checkbox"/></div><div><div> Editar</div><div> Copiar</div><div> Borrar</div></div></div> <td>1003</td> <td>Tangram</td> <td>10.5</td>	1003	Tangram	10.5

Solución

```
SELECT art.Codigo_art, art.Nombre_art, sum( lin.Cantidad_compra* lin.Precio_compra) as
importe_comprado
FROM articulos as art, lineas_compra as lin
WHERE art.Codigo_art=lin.Codigo_art
GROUP BY art.Codigo_art, art.Nombre_art
ORDER BY art.Codigo_art
```

### 3. CONSULTA 3

Actualizar el descuento actual de todos los artículos un 8% en los artículos de Decoración ya que hoy es 9 de octubre. Como se muestra en la figura.

✓ 3 filas afectadas. (La consulta tardó 0.0067 segundos.)

Solución 1:

```
update articulos, tipos, tipos_articulos
set articulos.descuento_actual=8
where articulos.Codigo_art=tipos_articulos.Codigo_art
and tipos_articulos.Id_tipo=tipos.Id_tipo
and tipos.Nombre_tipo='Decoración';
```

Solución 2 :

```
UPDATE articulos SET Descuento_actual = 8 WHERE Codigo_art in (SELECT Codigo_art FROM
tipos_articulos WHERE Id_tipo = 4004)
```

Solución 3:

```
UPDATE articulos
SET Descuento_actual=8
WHERE Codigo_art=
(SELECT articulos.Codigo_art
FROM tipos_articulos, tipos
WHERE articulos.Codigo_art= tipos_articulos.Codigo_art AND
tipos.Id_tipo=tipos_articulos.Id_tipo AND tipos.Nombre_tipo='Decoracion');
```

Solución 4:

```
UPDATE ARTICULOS SET descuento_actual = 8.0 WHERE codigo_art IN
(SELECT codigo_art FROM TIPOS_ARTICULOS WHERE id_tipo IN
(SELECT id_tipo FROM TIPOS WHERE nombre_tipo = 'decoración')
);
```

#### 4. CONSULTA 4

Obtener la consulta que muestre el código de artículo, la fecha de venta, nombre del artículo, el descuento, el precio con descuento de las ventas del artículo artículo cuya fecha sea 14/12/2019. Como se muestra en la imagen.

+ Opciones

Codigo_art	fecha_ven	nombre_art	precio_actual	descuento_actual	preciocondescuento
1001	2019-12-14	Muñeca	20.5	5	19.474999999999998
1002	2019-12-14	Coche de carreras	15.3	2	14.994000186920166
1003	2019-12-14	Tangram	30.5	1	30.195

Solución 1

```
select l.Codigo_art , v.fecha_ven, nombre_art, precio_actual, descuento_actual, precio_actual *
( 1 - descuento_actual/100) as preciocondescuento
from articulos a ,lineas_venta l, ventas v
where a.Codigo_art = l.Codigo_art and v.Id_venta=l.Id_venta AND fecha_ven='2019-12-14';
```

Solución 2

```
SELECT A.Codigo_art, V.Fecha_ven, A.Nombre_art, A.Precio_actual, A.Descuento_actual,
(A.Precio_actual-(A.Precio_actual*A.Descuento_actual/100)) AS preciocondescuento
FROM lineas_venta L, ventas V, articulos A
WHERE V.Fecha_ven='2019-12-14' AND V.Id_venta=L.Id_venta AND A.Codigo_art=L.Codigo_art
GROUP BY A.Codigo_art
ORDER BY L.Codigo_art;
```

## 5. CONSULTA 5

Actualizar el descuento del cliente que es el hermano del propietario con dni 123456789, al cual se le hace un descuento del 25% en el importe de su compra. Nos olvidemos que hay un descuento para los clientes y otro para los artículos. Como se muestra en la imagen:

✓ 1 fila afectada. (La consulta tardó 0.0074 segundos.)

Solución:

```
update clientes
set clientes.Descuento_venta = 25
where clientes.Dni='123456789';
```

## 6. CONSULTA 6

Mostrar el importe total de las ventas del año 2019, el importe total de las compras del año 2019 y el beneficio que se obtiene como el importe total de las ventas - importe total de las compras. Como se muestra en la imagen:

+ Opciones		
importe_ventas	importe_compras	beneficio
102.10000038146973	72.29999923706055	29.80000114440918

Solución 1:

```
create or replace view importe_ventas AS
select sum(vl.cantidad_venta * vl.Precio_venta ) as importe_ventas
from ventas as v, lineas_venta as vl
where v.Id_venta = vl.Id_venta;

create or replace view importe_compras AS
select sum(cl.Cantidad_compra * cl.Precio_compra ) as importe_compras
from compras as c, lineas_compra as cl
where c.id_compra = cl.id_compra;

select importe_ventas, importe_compras, (importe_ventas - importe_compras) as beneficio
from importe_ventas , importe_compras;
```

Solución 2:

```

SELECT SUM(Cantidad_venta * Precio_venta) as importe_ventas, (SELECT SUM(Cantidad_compra
* Precio_compra) FROM lineas_compra) as importe_compras,
(SUM(Cantidad_venta * Precio_venta) - (SELECT SUM(Cantidad_compra * Precio_compra) FROM
lineas_compra)) as beneficio
FROM lineas_venta

```

#### Solución 3:

```

select Total_compras, Total_ventas, total_ventas - total_compras as Beneficio
from (select sum(cantidad_compra*precio_compra) as total_compras
from lineas_compra) TOTALES_COMPRAS,
(select sum(cantidad_venta*Precio_venta) as Total_ventas
from lineas_venta) TOTALES_VENTAS;

```

#### Solución 4:

```

SELECT importe_ventas, importe_compras,
(importe_ventas-importe_compras) AS beneficio
FROM (SELECT sum(l.cantidad_venta*l.precio_venta) AS importe_ventas
FROM ventas v, lineas_venta l
WHERE v.id_venta=l.id_venta AND
v.fecha_ven BETWEEN '2019-01-01' AND '2019-12-31') tabla1,
(SELECT sum(l.cantidad_compra*l.precio_compra) AS importe_compras
FROM compras c, lineas_compra l
WHERE c.id_compra=l.id_compra AND
c.fecha_compra BETWEEN '2019-01-01' AND '2019-12-31') tabla2;

```

#### Solución 5:

```

SELECT importe_ventas,importe_compras,beneficio
FROM
(SELECT SUM(lineas_venta.Precio_venta*lineas_venta.Cantidad_venta) AS importe_ventas
FROM lineas_venta, ventas
WHERE lineas_venta.Id_venta=ventas.Id_venta AND ventas.Fecha_ven BETWEEN '2019-01-01'
AND '2019-12-31'
) importeventas,
(SELECT SUM(lineas_compra.Precio_compra*lineas_compra.Cantidad_compra) AS
importe_compras
FROM lineas_compra, compras
WHERE lineas_compra.Id_compra=compras.Id_compra AND compras.Fecha_compra BETWEEN
'2019-01-01' AND '2019-12-31'
)importcompras,
(SELECT importe_ventas- importe_compras AS beneficio
FROM
(SELECT SUM(lineas_venta.Precio_venta*lineas_venta.Cantidad_venta) AS importe_ventas

```



```
FROM lineas_venta, ventas
WHERE lineas_venta.Id_venta=ventas.Id_venta AND ventas.Fecha_ven BETWEEN '2019-01-01'
AND '2019-12-31') importeventas
,
(SELECT SUM(lineas_compra.Precio_compra*lineas_compra.Cantidad_compra) AS
importe_compras
FROM lineas_compra, compras
WHERE lineas_compra.Id_compra=compras.Id_compra AND compras.Fecha_compra
BETWEEN '2019-01-01' AND '2019-12-31'
) importecompras
) beneficio;
```

#### Solución 6:

```
SELECT
sum(LINEAS_VENTA.precio_venta * CAST(cantidad_venta AS int)) AS importe_ventas,
sum(LINEAS_COMPRA.precio_compra * CAST(cantidad_compra AS int)) AS importe_compras,
sum(LINEAS_VENTA.precio_venta * CAST(cantidad_venta AS int)) -
sum(LINEAS_COMPRA.precio_compra * CAST(cantidad_compra AS int)) AS beneficio
FROM LINEAS_COMPRA
INNER JOIN ARTICULOS ON LINEAS_COMPRA.codigo_art = ARTICULOS.codigo_art
INNER JOIN LINEAS_VENTA ON LINEAS_VENTA.codigo_art = ARTICULOS.codigo_art
INNER JOIN VENTAS ON LINEAS_VENTA.id_venta = VENTAS.id_venta
INNER JOIN COMPRAS ON LINEAS_COMPRA.id_compra = COMPRAS.id_compra
WHERE
EXTRACT(YEAR FROM VENTAS.fecha_ven) = 2019 AND
EXTRACT(YEAR FROM COMPRAS.fecha_compra) = 2019;
```