



Tecnicatura Universitaria
en Programación

LABORATORIO DE COMPUTACIÓN II

Unidad Temática N°2:
Subconsultas

Guía
1° Año – 2° Cuatrimestre

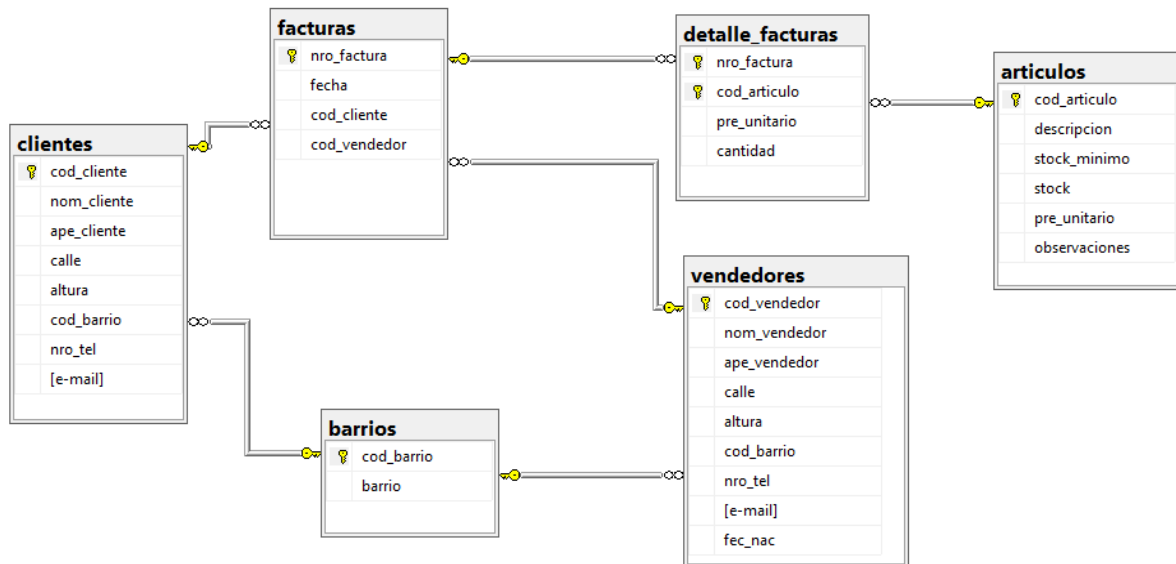


Índice

Problema 2.1: Subconsultas en el Where	2
Problema 2.2: Subconsultas en el Having.....	6
Problema 2.3: Otras Subconsultas.....	9
BIBLIOGRAFÍA	12

Problema 2.1: Subconsultas en el Where

Continuando con el negocio dedicado a la venta mayorista de artículos de librería se vuelve a presentar el diagrama de la base de datos “LIBRERIA”.



Guía 2 Página 2 Ejercicio 1 Resultados 23

Se solicita un listado de artículos cuyo precio es inferior al promedio de precios de todos los artículos.

```
Select cod_articulo, descripcion, pre_unitario
from articulos
where pre_unitario < (Select AVG (pre_unitario) from articulos)
```

Guía 2 Página 2 Ejercicio 2 Resultados 9

Emitir un listado de los artículos que no fueron vendidos este año. En ese listado solo incluir aquellos cuyo precio unitario del artículo oscile entre 50 y 100.

```
SELECT a.cod_articulo, a.pre_unitario, descripcion 'Articulo'
FROM articulos a
WHERE a.pre_unitario between 1.00 AND 1000.00
and a.cod_articulo NOT IN ( SELECT distinct df1.cod_articulo
                           FROM facturas f1, detalle_facturas df1
                           WHERE YEAR(f1.fecha) != YEAR(getdate())
                           and df1.nro_factura = f1.nro_factura )
```

Guía 2 Página 2 Ejercicio 3 Resultados 9

Genere un reporte con los clientes que vinieron más de 2 veces el año pasado.

```
Select ape_cliente + ' ' + nom_cliente 'CLIENTE'
from clientes c
join facturas f on f.cod_cliente = c.cod_cliente
where YEAR(fecha)= YEAR(GETDATE())-1
group by ape_cliente , nom_cliente
having count(nro_factura) > 2
```

Guía 2 Página 2 Ejercicio 4 Resultados 4

Se quiere saber qué clientes no vinieron entre el 12/12/2015 y el 13/7/2020

```
select cod_cliente, ape_cliente+' '+nom_cliente Clientes
from clientes c
where not exists (select cod_cliente
from facturas f
where f.cod_cliente=c.cod_cliente
and fecha not between '2015-12-12' and '2020-07-16')
```

Guía 2 Página 2 Ejercicio 5 Resultados 11

Genere un reporte con los clientes que vinieron más de 2 veces el año pasado.

```
SELECT c.ape_cliente + ', ' + c.nom_cliente 'Clientes', count(f.cod_cliente)
FROM clientes c, facturas f
WHERE c.cod_cliente = f.cod_cliente and year(f.fecha) = 2022
group by c.ape_cliente + ', ' + c.nom_cliente
having count( f.cod_cliente) > 2
```

Guía 2 Página 2 Ejercicio 6 Resultados 35

Mostrar los datos de las facturas para los casos en que por año se hayan hecho menos de 9 facturas.

```
select f.nro_factura 'Factura', fecha 'Fecha',
c.ape_cliente+space(1)+c.nom_cliente 'Cliente',
v.ape_vendedor+space(1)+v.nom_vendedor 'Vendedor' from facturas f
join clientes c on c.cod_cliente=f.cod_cliente
join vendedores v on v.cod_vendedor=f.cod_vendedor
```

```
where year(fecha) in (select YEAR(fecha) from facturas group by YEAR(fecha) having
COUNT(nro_factura)<9)
```

Guía 2 Página 3 Ejercicio 7 Resultados 119

Emitir un reporte con las facturas cuyo importe total haya sido superior a 1.500 (incluir en el reporte los datos de los artículos vendidos y los importes).

```
select f.nro_factura 'Factura', f.fecha 'Fecha', c.ape_cliente+space(1)+c.nom_cliente
'Cliente',
v.ape_vendedor+space(1)+v.nom_vendedor 'Vendedor', sum(df.cantidad * df.pre_unitario)
from facturas f
join detalle_facturas df on df.nro_factura = f.nro_factura
join clientes c on c.cod_cliente=f.cod_cliente
join vendedores v on v.cod_vendedor=f.cod_vendedor
group by f.nro_factura, f.fecha, c.ape_cliente+space(1)+c.nom_cliente,
v.ape_vendedor+space(1)+v.nom_vendedor
having sum(df.cantidad * df.pre_unitario) > 1500
```

Guía 2 Página 3 Ejercicio 8 Resultados 5

Se quiere saber qué vendedores nunca atendieron a estos clientes: 1 y 6. Muestre solamente el nombre del vendedor.

```
select distinct V.nom_vendedor Vendedor
from facturas F
join vendedores V on V.cod_vendedor = F.cod_vendedor
where cod_cliente in (select cod_cliente
from clientes where cod_cliente not in (1, 6))
```

Guía 2 Página 3 Ejercicio 9 Resultados 23

Listar los datos de los artículos que superaron el promedio del Importe de ventas de \$ 1.000.

```
select distinct a.cod_articulo, descripcion, a.pre_unitario
from articulos a
join detalle_facturas df on df.cod_articulo = a.cod_articulo
where 1000 < (select (sum(pre_unitario*cantidad)/count(cod_articulo))
from detalle_facturas)
order by 1
```

Guía 2 Página 3 Ejercicio 10 Resultados 9

¿Qué artículos nunca se vendieron? Tenga además en cuenta que su nombre comience con letras que van de la “d” a la “p”. Muestre solamente la descripción del artículo.

```
select a.descripcion
from articulos a
where a.cod_articulo NOT IN (select distinct cod_articulo from detalle_facturas)
```

Guía 2 Página 3 Ejercicio 11 Resultados 3

Listar número de factura, fecha y cliente para los casos en que ese cliente haya sido atendido alguna vez por el vendedor de código 3.

```
select nro_factura, fecha, c.nom_cliente+' '+c.ape_cliente 'Nombre Completo'
from facturas f join clientes c on c.cod_cliente=f.cod_cliente
where 3 not in (select distinct cod_vendedor
               from facturas f
               where c.cod_cliente = f.cod_cliente)
```

Guía 2 Página 3 Ejercicio 12 Resultados 37

Listar número de factura, fecha, artículo, cantidad e importe para los casos en que todas las cantidades (de unidades vendidas de cada artículo) de esa factura sean superiores a 40.

```
select df.nro_factura as 'numero de factura',
       f.fecha as 'Fecha',
       df.cod_articulo as 'cod articulo',
       df.cantidad as 'Cantidad de unidades',
       sum(df.cantidad*df.pre_unitario) as 'Importe'
from facturas f inner join detalle_facturas df on df.nro_factura =
f.nro_factura
       inner join articulos a on df.cod_articulo =
a.cod_articulo
where exists(select df.nro_factura
            from articulos
            where df.cantidad > 40)
group by df.cod_articulo, f.fecha, df.nro_factura, df.cantidad
```

Guía 2 Página 3 Ejercicio 13 Resultados 5

Emitir un listado que muestre número de factura, fecha, artículo, cantidad e importe; para los casos en que la cantidad total de unidades vendidas sean superior a 80.

```
select df.nro_factura as 'numero de factura',
       f.fecha as 'Fecha',
       df.cod_articulo as 'cod articulo',
       df.cantidad as 'Cantidad de unidades',
       sum(df.cantidad*df.pre_unitario) as 'Importe'
```

```
from facturas f inner join detalle_facturas df on df.nro_factura =  
f.nro_factura  
inner join articulos a on df.cod_articulo =  
a.cod_articulo  
where df.cantidad > 50  
group by df.cod_articulo, f.fecha, df.nro_factura, df.cantidad
```

Guía 2 Página 3 Ejercicio 14 Resultados 904

Realizar un listado de número de factura, fecha, cliente, artículo e importe para los casos en que al menos uno de los importes de esa factura sea menor a 3.000.

```
select f.nro_factura, f.fecha, c.nom_cliente + ' ' + c.ape_cliente as  
'Cliente', a.descripcion, SUM(d.pre_unitario*d.cantidad) 'Total'  
from facturas f join clientes c on f.cod_cliente = c.cod_cliente  
join detalle_facturas d on d.nro_factura = f.nro_factura  
join articulos a on d.cod_articulo = a.cod_articulo  
where (select SUM(d.pre_unitario*d.cantidad) from detalle_facturas d) > 3000  
group by c.nom_cliente + ' ' + c.ape_cliente, f.nro_factura, f.fecha, a.descripcion  
having SUM(d.pre_unitario * d.cantidad) between 0 and 3000  
ORDER BY TOTAL DESC
```

Problema 2.2: Subconsultas en el Having

Guía 2 Página 4 Ejercicio 1 Resultados 2

Se quiere saber ¿cuándo realizó su primer venta cada vendedor? y ¿cuánto fue el importe total de las ventas que ha realizado? Mostrar estos datos en un listado solo para los casos en que su importe promedio de vendido sea superior al importe promedio general (importe promedio de todas las facturas).

```
select v.nom_vendedor + ' , ' +v.ape_vendedor 'nombre  
completo',sum(pre_unitario*cantidad) 'total',min(fecha)'fecha de primer  
venta'  
from facturas f join detalle_facturas df on
```

```
f.nro_factura=df.nro_factura
join vendedores v on v.cod_vendedor=f.cod_vendedor
group by v.nom_vendedor,v.ape_vendedor
having avg(pre_unitario*cantidad)< (select
avg(pre_unitario*cantidad)from detalle_facturas)
```

Guía 2 Página 4 Ejercicio 2 Resultados 3

Liste los montos totales mensuales facturados por cliente y además del promedio de ese monto y el promedio de precio de artículos Todos esto datos correspondientes a período que va desde el 1° de febrero al 30 de agosto del 2014. Sólo muestre los datos si esos montos totales son superiores o iguales al promedio global.

```
select c.cod_cliente, c.nom_cliente+SPACE(1)+c.ape_cliente 'Cliente',
month(fecha) 'Mes', sum(pre_unitario*cantidad) 'Monto total',
sum(pre_unitario*cantidad)/COUNT(f.nro_factura) 'Promedio monto total',
AVG(pre_unitario) 'Promedio precio artículos'
from facturas f join detalle_facturas df on f.nro_factura =
df.nro_factura
join clientes c on f.cod_cliente = c.cod_cliente
where fecha between cast( '02/01/2014' as date) and cast( '08/30/2017' as date)
group by c.cod_cliente, c.nom_cliente+SPACE(1)+c.ape_cliente,
MONTH(fecha)
having sum(pre_unitario*cantidad) >= (select AVG(pre_unitario*cantidad)
from detalle_facturas)
```

Guía 2 Página 4 Ejercicio 3 Resultados 1

Por cada artículo que se tiene a la venta, se quiere saber el importe promedio vendido, la cantidad total vendida por artículo, para los casos en que los números de factura no sean uno de los siguientes: 2, 10, 7, 13, 22 y que ese importe promedio sea inferior al importe promedio de ese artículo.

```
SELECT a.cod_articulo 'Codigo',descripcion 'Articulo',
        AVG(cantidad * df.pre_unitario) 'Importe promedio vendido',
        SUM(cantidad) 'Cantidades vendidas'
from articulos a
join detalle_facturas df on a.cod_articulo = df.cod_articulo
where nro_factura not in (2,10,7,13,22)
group by a.cod_articulo,descripcion
having AVG(cantidad * df.pre_unitario) < (SELECT AVG(df2.cantidad *
df2.pre_unitario) from detalle_facturas df2 where df2.cod_articulo =
a.cod_articulo )
```

Guía 2 Página 4 Ejercicio 4 Resultados 300

Listar la cantidad total vendida, el importe y promedio vendido por fecha, siempre que esa cantidad sea superior al promedio de la cantidad global.
Rotule y ordene.


```
select sum(cantidad) 'Cantidad total vendida', sum(cantidad*pre_unitario) 'Importe',
avg(cantidad) 'Promedio Vendido', f.fecha Fecha
from detalle_facturas d
join facturas f on f.nro_factura =d.nro_factura
group by f.fecha
having sum(cantidad) > (select avg(cantidad) from detalle_facturas)
order by f.fecha desc
```

Guía 2 Página 4 Ejercicio 5 Resultados 19

Se quiere saber el promedio del importe vendido y la fecha de la primer venta por fecha y artículo para los casos en que las cantidades vendidas oscilen entre 5 y 20 y que ese importe sea superior al importe promedio de ese artículo.

```
SELECT c.ape_cliente + ', ' + c.nom_cliente + ' (' + CAST(c.cod_cliente
AS varchar) + ')' AS 'NOMBRE',
       c.calle + ' ' + CAST(c.altura AS varchar) 'DIRECCION',
       UPPER(LEFT(b.barrio, 1)) + LOWER(RIGHT(b.barrio, LEN(b.barrio)-1))
'BARRIO', 'Cliente' AS 'INTEGRANTE'
FROM clientes c JOIN barrios b ON c.cod_barrio = b.cod_barrio
WHERE (c.calle IS NOT NULL AND c.altura IS NOT NULL)
UNION
SELECT v.ape_vendedor + ', ' + v.nom_vendedor + ' (' +
CAST(v.cod_vendedor AS varchar) + ')' 'NOMBRE',
       v.calle + ' ' + CAST(v.altura AS varchar) 'DIRECCION',
       UPPER(LEFT(b.barrio, 1)) + LOWER(RIGHT(b.barrio, LEN(b.barrio)-1))
'BARRIO', 'Vendedor' AS 'INTEGRANTE'
FROM vendedores v JOIN barrios b ON v.cod_barrio = b.cod_barrio
WHERE v.cod_vendedor BETWEEN 3 AND 6 AND (v.calle IS NOT NULL AND v.altura IS NOT
NULL)
ORDER BY 1, 4
```

Guía 2 Página 4 Ejercicio 6 Resultados 336

Emita un listado con los montos diarios facturados que sean inferior al importe promedio general.

```
select f.fecha, sum(cantidad*pre_unitario)'IMPORTE'
from facturas f join detalle_facturas df on
f.nro_factura=df.nro_factura
group by f.fecha
having avg(cantidad*pre_unitario) <
(SELECT avg(cantidad * pre_unitario) from detalle_facturas)
```

Guía 2 Página 4 Ejercicio 7 Resultados 7

Se quiere saber la fecha de la primera y última venta, el importe total facturado por cliente para los años que oscilen entre el 2010 y 2015 y que el importe promedio facturado sea menor que el importe promedio total para ese cliente.

```
select c.cod_cliente Codigo, c.ape_cliente+' '+c.nom_cliente Cliente, min(f.fecha)
'Fecha 1º venta', max(f.fecha) 'Fecha ult. venta',
sum(d.cantidad*d.pre_unitario) 'Total Facturado'
from facturas f
join clientes c on f.cod_cliente = c.cod_cliente
```

```
join detalle_facturas d on d.nro_factura = f.nro_factura
where year(f.fecha) between 2010 and 2015
group by c.cod_cliente, c.ape_cliente+' '+c.nom_cliente
having sum(d.cantidad*d.pre_unitario) < (select
sum(d.cantidad*d.pre_unitario)/count(distinct d.nro_factura)
from facturas f
join detalle_facturas d on d.nro_factura = f.nro_factura)
order by 1
```

Guía 2 Página 5 Ejercicio 8 Resultados 2

Realice un informe que muestre cuánto fue el total anual facturado por cada vendedor, para los casos en que el nombre de vendedor no comience con 'B' ni con 'M', que los números de facturas oscilen entre 5 y 25 y que el promedio del monto facturado sea inferior al promedio de ese año.

```
select v.cod_vendedor Codigo, v.ape_vendedor+' '+v.nom_vendedor Vendedor,
sum(d.cantidad*d.pre_unitario) 'Total Facturado', v.nom_vendedor
from facturas f
join vendedores v on v.cod_vendedor = f.cod_vendedor
join detalle_facturas d on d.nro_factura = f.nro_factura
where v.nom_vendedor NOT LIKE 'M%' and v.nom_vendedor NOT LIKE 'B%'
group by v.cod_vendedor, v.ape_vendedor+' '+v.nom_vendedor, v.nom_vendedor
having avg(d.cantidad*d.pre_unitario) < (select
sum(d.cantidad*d.pre_unitario)/count(distinct d.nro_factura)
from facturas f
join detalle_facturas d on d.nro_factura = f.nro_factura)
order by 1
```

Problema 2.3: Otras Subconsultas

Guía 2 Página 6 Ejercicio 1 Resultados 32

Se quiere listar el precio de los artículos y la diferencia de éste con el precio del artículo más caro:

```
Select cod_articulo 'Codigo',
       descripcion,
       pre_unitario 'Precio',
       (Select max(pre_unitario) from articulos) - pre_unitario 'Diferencia'
from articulos
order by 2 desc
```

Guía 2 Página 6 Ejercicio 2 Resultados 23

Listar el precio actual de los artículos y el precio histórico vendido más barato

```
SELECT descripcion 'ARTICULO', a.pre_unitario 'PRECIO ACTUAL',
(SELECT MIN(d.pre_unitario)
FROM detalle_facturas D
JOIN articulos A ON A.cod_articulo = D.cod_articulo
JOIN facturas F ON F.nro_factura = D.nro_factura
WHERE D.cod_articulo = A.cod_articulo) 'PRECIO HISTORICO'
FROM articulos a
join detalle_facturas d on d.cod_articulo = A.cod_articulo
```

GROUP BY descripcion , a.pre_unitario

Guía 2 Página 6 Ejercicio 3 Resultados 130

Se quiere emitir un listado de las facturas del año en curso detallando número de factura, cliente, fecha y total de la misma.

```
select df.nro_factura as 'num de factura',
       c.nom_cliente + '-' + c.ape_cliente as 'Nom cliente',
       str(day(f.fecha)) + '-' + str(month(f.fecha)) + '-' + str(year(f.fecha)) as
'fecha',
       sum(df.pre_unitario*df.cantidad) as 'total'
from facturas f inner join detalle_facturas df on df.nro_factura =
f.nro_factura
       inner join clientes c on c.cod_cliente = f.cod_cliente
where year(f.fecha) = year(GETDATE())
group by df.nro_factura, c.nom_cliente, c.ape_cliente, f.fecha
```

Guía 2 Página 6 Ejercicio 4 Resultados 32

Emitir un listado con el código y descripción de los artículos su precio actual, el precio promedio al cuál se vendió el año pasado (ver diferencia entre el promedio ponderado y el promedio simple)

```
select a.cod_articulo, a.descripcion, a.pre_unitario,
(select avg(df.pre_unitario) from detalle_facturas df , facturas f where
df.nro_factura=f.nro_factura and year(f.fecha)=2021
and df.cod_articulo = a.cod_articulo) 'precio promedio 2021'
from articulos a
```

Guía 2 Página 6 Ejercicio 5 Resultados 32

Generar un reporte un listado con la código y descripción de los artículos su precio actual, el precio más barato y el más caro al que se vendió hace 5 años.

```
select a.cod_articulo, a.descripcion, a.pre_unitario,
(select min(df.pre_unitario) from detalle_facturas df , facturas f where
df.nro_factura=f.nro_factura and year(f.fecha) between 2017 and 2021
and df.cod_articulo = a.cod_articulo) 'precio mas barato 5 años',
(select max(df.pre_unitario) from detalle_facturas df , facturas f where
df.nro_factura=f.nro_factura and year(f.fecha) between 2017 and 2021
and df.cod_articulo = a.cod_articulo) 'precio mas caro 5 años'
from articulos a
```

Guía 2 Página 6 Ejercicio 6 Resultados 2

Descontar un 3,5% los precios de los artículos que se vendieron menos de 5 unidades los últimos 3 meses.

```
select a.descripcion,
       sum(cantidad) AS 'cantidad de ventas',
       f.fecha AS 'fecha',
       a.pre_unitario AS 'precio',
       a.pre_unitario * 0.965 AS 'precio descontado'
from detalle_facturas df
     join facturas f on df.nro_factura = f.nro_factura
     join articulos a on df.cod_articulo = a.cod_articulo
where datediff(month,f.fecha, getdate()) between 0 and 2
group by a.descripcion,f.fecha,a.pre_unitario
having sum(cantidad) < 6
```

Guía 2 Página 6 Ejercicio 7 Resultados 4

Se quiere eliminar los clientes que no vinieron nunca.

```
select *
from clientes
where cod_cliente not in (select distinct cod_cliente
                           from facturas)

delete clientes
from clientes
where cod_cliente not in (select distinct cod_cliente
                           from facturas)
```

Guía 2 Página 6 Ejercicio 8 Resultados 4

Eliminar los clientes que hace más de 10 años que no vienen.

```
--delete clientes
select c.cod_cliente, c.ape_cliente + ' ' + c.nom_cliente 'Clientes'
from clientes c
where c.cod_cliente not in (select f.cod_cliente
                           from facturas f
                           join clientes c on c.cod_cliente =
f.cod_cliente
                           where fecha between cast('08-09-2012'
as date) AND cast('08-09-2022' as date))
```

BIBLIOGRAFÍA

Gorman K., Hirt A., Noderer D., Rowland-Jones J., Sirpal A., Ryan D. & Woody B (2019) Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud. Packt Publishing Ltd. Birmingham UK

Microsoft (2021) SQL Server technical documentation. Disponible en: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

Opel, A. & Sheldon, R. (2010). Fundamentos de SQL. Madrid. Editorial Mc Graw Hill

Varga S., Cherry D., D'Antoni J. (2016). Introducing Microsoft SQL Server 2016 Mission-Critical Applications, Deeper Insights, Hyperscale Cloud. Washington. Microsoft Press



Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera: Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.