



Tecnicatura Universitaria
en Programación

LABORATORIO DE COMPUTACIÓN II

Unidad Temática N°2:
Subconsultas

Guía
1° Año – 2° Cuatrimestre

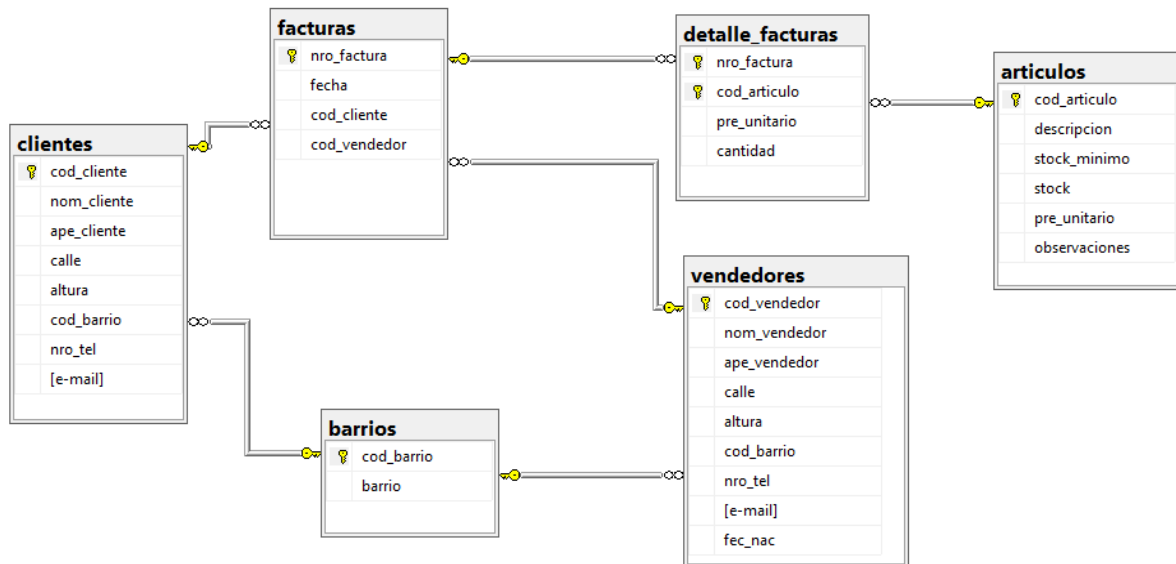


Índice

Problema 2.1: Subconsultas en el Where	2
Problema 2.2: Subconsultas en el Having.....	4
Problema 2.3: Otras Subconsultas.....	6
BIBLIOGRAFÍA	7

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”.



```

/*3.Genere un reporte con los clientes que vinieron más de 2 veces el año
pasado. */
select c.cod_cliente 'Codigo Cliente',
c.ape_cliente +', '+c.nom_cliente as 'Nombre Completo'
from clientes c
where exists (select cod_cliente
from facturas f
where year(fecha) = year(getdate())-1
and f.cod_cliente=c.cod_cliente
group by cod_cliente
having count(*) > 2)

--otra solucion
select c.cod_cliente 'Codigo Cliente',c.ape_cliente +', '+c.nom_cliente as 'Nombre
Completo'
from clientes c
where 2 <(select count(*)
from facturas f
where year(fecha) = year(getdate())-1
and f.cod_cliente=c.cod_cliente)

set dateformat dmy
--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 Cliente
from clientes as c
where not exists (select cod_cliente
from facturas as f
where (f.cod_cliente=c.cod_cliente) and
(fecha between '12/12/2015' and '13/7/2020'))

--5. Listar los datos de las facturas de los clientes que solo vienen a comprar
  
```

```
--en febrero es decir que todas las veces que vienen a comprar haya sido
--en el mes de febrero (y no otro mes).
select f.nro_factura, format(fecha, 'dd/MM/yyyy') 'Fecha', c.cod_cliente,
ape_cliente+' '+nom_cliente Cliente
from facturas f join clientes c on f.cod_cliente=c.cod_cliente
where 2=all(select month(fecha)
from facturas f1
where f1.cod_cliente = c.cod_cliente)

--6. Mostrar los datos de las facturas para los casos en que por año se hayan
--hecho menos de 9 facturas.
select f.nro_factura 'Número de factura', format(fecha, 'yyyy-MM-dd') Fecha,
      nom_cliente + ' ' + ape_cliente Cliente,
      nom_vendedor + ' ' + ape_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)
order by 2
--otra solución
select f.nro_factura 'Número de factura', format(fecha, 'yyyy-MM-dd') Fecha,
      nom_cliente + ' ' + ape_cliente Cliente,
      nom_vendedor + ' ' + ape_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 9>(select count(*)
from facturas f1
where year(fecha)=year(f.fecha)
)

--7. 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, FORMAT(fecha, 'dd/MM/yyyy') 'Fecha',
(cantidad * pre_unitario) 'Importe'
from facturas f join detalle_facturas df on df.nro_factura = f.nro_factura
where 1500 <(select sum(cantidad * pre_unitario)
from detalle_facturas d1
where d1.nro_factura=f.nro_factura)

--8. Se quiere saber qué vendedores nunca atendieron a estos clientes: 1 y 6.
--Muestre solamente el nombre del vendedor.
select ape_vendedor+' '+nom_vendedor Vendedor,
calle+' '+cast(altura as varchar) Direccion,
nro_tel Telefono, [e-mail] Email
from vendedores v
where not exists(select nro_factura
from facturas
where cod_vendedor = v.cod_vendedor and cod_cliente in (1,6))

--9. Listar los datos de los artículos que superaron el promedio del Importe de
--ventas de $ 1.000
select a.cod_articulo, descripcion
from articulos a
where 1000 <(select avg(cantidad*pre_unitario)
from detalle_facturas d
where a.cod_articulo=d.cod_articulo)

--10. ¿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 descripcion as 'Articulo'
from articulos a
where not exists(select df.cod_articulo
from detalle_facturas df
where a.cod_articulo=df.cod_articulo)
AND descripcion like '[d-p]%'

--11. 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.cod_cliente, ape_cliente+', '+nom_cliente Cliente
from facturas as f join clientes as c on f.cod_cliente=c.cod_cliente
where 3=any(select cod_vendedor
from facturas as f1
where f1.cod_cliente=f.cod_cliente)

--12. 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 f.nro_factura, fecha, descripcion, cantidad, cantidad*d.pre_unitario importe
from detalle_facturas d join facturas f on d.nro_factura=f.nro_factura
join articulos a on a.cod_articulo=d.cod_articulo
where 40<all(select cantidad
from detalle_facturas d1
where d1.nro_factura=f.nro_factura)

--13. Emitir un listado que muestre número de factura, fecha, artículo, cantidad
--e importe; para los casos en que la cantidad total de unidades vendidas en esa
--factura sea superior a 80.
select f.nro_factura, fecha, descripcion, cantidad, cantidad*d.pre_unitario importe
from detalle_facturas d join facturas f on d.nro_factura=f.nro_factura
join articulos a on a.cod_articulo=d.cod_articulo
where 80<(select sum(cantidad)
from detalle_facturas d1
where d1.nro_factura=f.nro_factura)

--14. Realizar un listado de número de factura, fecha, cliente, artículo e importe
--para los casos en que al menos uno de los importes (de los detalles de factura)
--de esa factura sea menor a 3.000.
select f.nro_factura, fecha, descripcion, cantidad, cantidad*d.pre_unitario importe
from detalle_facturas d join facturas f on d.nro_factura=f.nro_factura
join articulos a on a.cod_articulo=d.cod_articulo
where 3000>any(select cantidad*pre_unitario
from detalle_facturas d1
where d1.nro_factura=f.nro_factura)

```

Problema 2.2: Subconsultas en el Having

```

--1. 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.cod_vendedor, ape_vendedor+', '+nom_vendedor vendedor,
min(fecha)'1er.vta.', sum(cantidad*pre_unitario)'importe total'
from facturas f join detalle_facturas d on f.nro_factura=d.nro_factura
join vendedores v on v.cod_vendedor=f.cod_vendedor

```

```
group by v.cod_vendedor,ape_vendedor+' '+nom_vendedor
having sum(cantidad*pre_unitario)/count(distinct f.nro_factura)
>(select sum(cantidad*pre_unitario)/count(distinct nro_factura)
  from detalle_facturas)

--3-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,sum(cantidad*d.pre_unitario)/count(distinct d.nro_factura)
promedio,
sum(cantidad) 'cant. total'
from facturas f join detalle_facturas d on f.nro_factura=d.nro_factura
join articulos a on a.cod_articulo=d.cod_articulo
where f.nro_fatura not in (2, 10, 7, 13,22)
group by a.cod_articulo
having sum(cantidad*d.pre_unitario)/count(distinct d.nro_factura)<
(select sum(cantidad*pre_unitario)/count(distinct d.nro_factura)
from detalle_facturas d1
where d1.cod_articulo=a.cod_articulo)

--4-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 fecha,sum(cantidad) 'cant. total',sum(cantidad*d.pre_unitario) importe
sum(cantidad*d.pre_unitario)/count(distinct d.nro_factura) promedio
from detalle_facturas d join facturas f on f.nro_factura=d.nro_factura
group by fecha
having sum(cantidad)>(select avg(cantidad) from detalle_facturas)

--5-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 el importe total (por fecha y artículo)
-- sea superior al importe promedio
--de ese artículo.
select fecha,d.cod_articulo,
sum(cantidad*d.pre_unitario)/count(distinct d.nro_factura) promedio,
min(fecha) '1er.vta'
from detalle_facturas d join facturas f on f.nro_factura=d.nro_factura
where cantidad between 5 and 20
group by fecha,d.cod_articulo
having sum(cantidad*d.pre_unitario)>
(select sum(cantidad*d1.pre_unitario)/count(distinct d1.nro_factura)
from detalle_facturas d1
where d1.cod_articulo=d.cod_articulo)
```

Problema 2.3: Otras Subconsultas

1. Se quiere listar el precio de los artículos y la diferencia de éste con el precio del artículo más caro:
2. Listar el precio actual de los artículos y el precio histórico vendido más barato
3. Se quiere emitir un listado de las facturas del año en curso detallando número de factura, cliente, fecha y total de la misma.
4. Emitir un listado con la 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)
5. 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.
6. Descontar un 3,5% los precios de los artículos que se vendieron menos de 5 unidades los últimos 3 meses.
7. Se quiere eliminar los clientes que no vinieron nunca.
8. Eliminar los clientes que hace más de 10 años que no vienen

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.