UNIVERSIDAD-TECNOLÓGICA-NACIONAL

F A C U L T A D - R E G I O N A L – C Ó R D O B A



*TECNICATURA UNIVERSITRA EN PROGRAMACIÓN* Materia: Laboratorio De Computación II

**COMISÍON: 1W1**

**GRUPO 5**

112674 – García Niz Lautaro

110643 – Ricatti Facundo

112658 – Giampaoli Ezequiel Maximiliano

112957 – González Tulián Noelia

110110 – López Rubén

**4.-Sistema de Venta de una Farmacéutica** Una cadena farmacéutica necesita llevar la información de su operatoria diaria de ventas al público y contar con reportes adecuados para un funcionamiento eficaz.

La farmacéutica tiene distintas sucursales dispersas a lo largo del país, las cuales son abastecidas con suministros a través de empresas de logística y distribución ajenas al sistema. Cuando se entregan suministros en alguna sucursal por parte de la empresa de logística se registra la cantidad de cada uno de ellos identificados por su código de barras, además de quién realizó dicha entrega por parte de la empresa y el responsable (empleado de la farmacéutica) que recibió los suministros.

La farmacéutica clasifica a los suministros en base a si son de venta libre o requieren una autorización o receta para poder ser entregados a los clientes. Los suministros pueden ser de diversos tipos como por ejemplo medicamentos, artículos de limpieza, higiene personal, estética, etc.

Cuando un cliente va a realizar compras en alguna sucursal, puede llevar algunos suministros que tengan cobertura por obra social (independientemente que sean de venta libre o no) por lo que deberá presentar la receta provista por el médico. Cada uno de esos productos puede o no tener algún descuento para dicha obra social en ese momento para la localidad donde se encuentra. Todos los meses las obras sociales envían la información de los descuentos que se deben ofrecer para cada tipo de producto para cada localidad del país.

Las obras sociales proveen un sistema externo de validación de afiliados, por lo que la farmacia debe ingresar los datos del cliente y el o los suministros que se encuentren en la receta además del nombre del médico y su matrícula. Una vez confirmados esos datos, estos sistemas externos devuelven un código de autorización a la farmacia que indica si la solicitud está aprobada o rechazada. A pesar de que este sistema no forma parte de la farmacéutica, se lleva un respaldo de la información ingresada en el portal de solicitud de coberturas por cuestiones de auditoría.

Todos estos descuentos realizados en suministros son facturados a la obra social que corresponda, por lo que se realiza un reporte con el listado de todos los tipos de suministros y las respectivas cantidades entregadas a los clientes para recibir el reembolso de la obra social. Una vez que este lote de suministros con descuentos ya fue reembolsado por la obra social se los marca para no ser vueltos a listar. Esto es así porque no necesariamente el listado para reembolso incluye suministros que hayan sido entregados en el mes que se está facturando.

Algunos posibles reportes que puede necesitar la farmacéutica son:

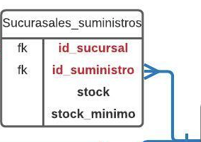
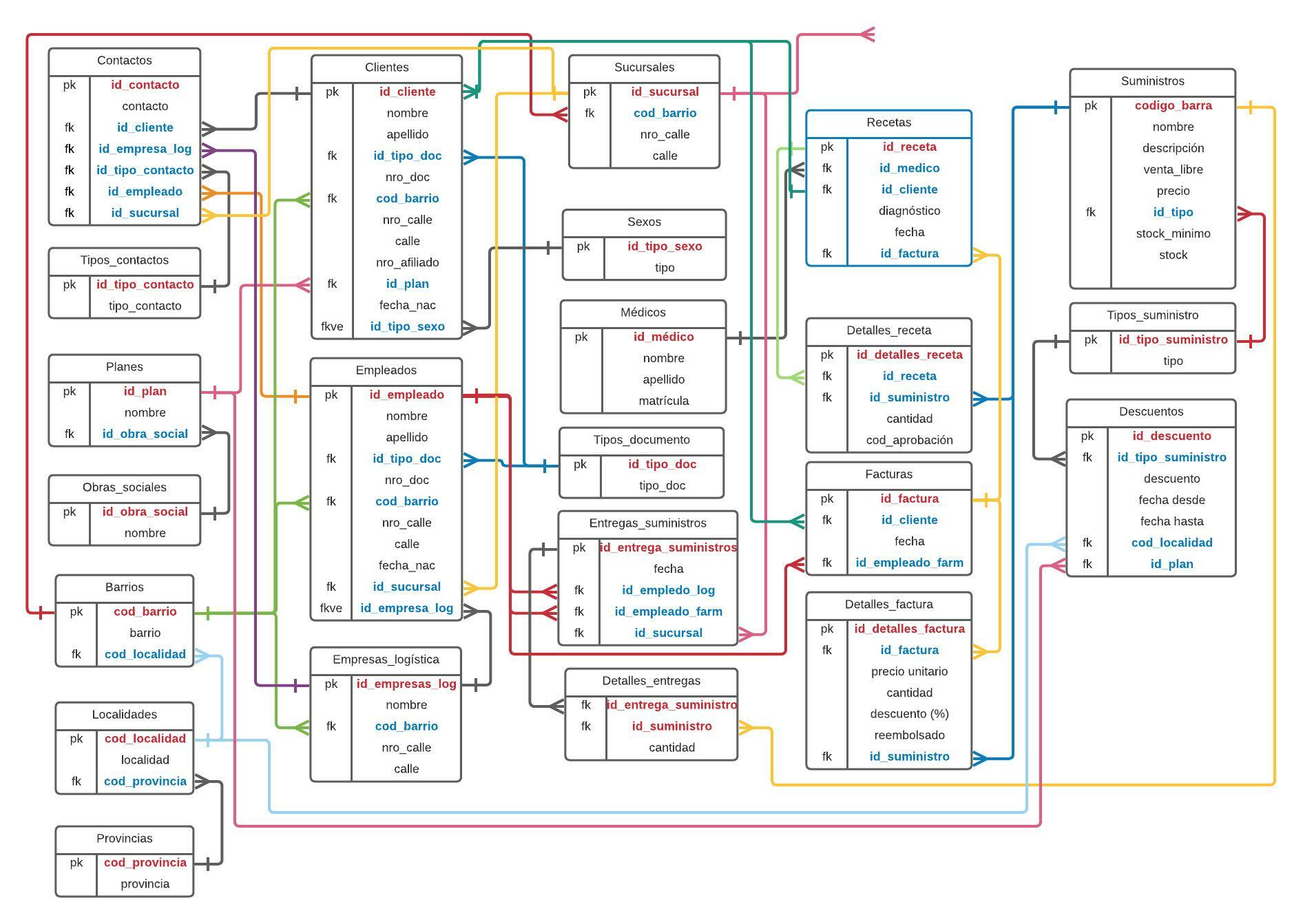
• Totales de facturación. Podría discriminarse por tipos de suministros, si fueron o no autorizados por obra social, etc.

• Reporte de reembolso para las obras sociales

• Clientes que realizaron compras el año actual sólo de suministros que requirieron autorización de obra social

• Cantidades de afiliados por obra social que realizaron compras, y total de descuentos aplicados

• Empresas de logística que más volumen de mercadería entrega



--1) Totales y promedio de facturación mensual con filtros (rango de fechas, tipo de suministros, autorizados por obra social)

create proc pa\_total\_facturacion

@fecha\_desde datetime = null,

@fecha\_hasta datetime = null,

@tipo varchar (50)= '%'

as

begin

select month (f.fecha) 'mes',

year (f.fecha) 'año',

s.nombre 'suministro',

t.tipo 'tipo de suministro',

format(sum (d.precio\_unitario\*d.cantidad-(d.precio\_unitario\*d.cantidad\*(1-descuento))),'c2','es-ar')'total de facturacion',

format(sum(d.precio\_unitario\*d.cantidad-(d.precio\_unitario\*d.cantidad\*(1-descuento)))/ count(f.id\_factura),'c2','es-ar') 'promedio de facturacion'

from facturas f join detalles\_factura d

on f.id\_factura = d.id\_factura

join suministros s on s.codigo\_barra=d.codigo\_barra

join tipos\_suministro t on t.id\_tipo\_suministro=s.id\_tipo\_suministro

join detalles\_receta dr on dr.codigo\_barra=s.codigo\_barra

where f.fecha between @fecha\_desde and @fecha\_hasta

and t.tipo like '%'+@tipo+'%'

and cod\_aprobacion is not null

group by month (f.fecha), year (f.fecha), s.nombre,t.tipo

order by 2,1

end

--exec pa\_total\_facturacion '1/1/2021','31/12/2021','paracetamol'

--2) Cantidad de clientes por mes, en cierto año, pasado por parametro y promedio de gastos

create proc pa\_clientes\_mes

@anio int

as

begin

select month(fecha) 'mes',

year(fecha) 'año',

count(f.id\_cliente) 'cantidad',

format(sum(df.precio\_unitario\*cantidad\*(1-descuento)) / count(f.id\_cliente),'c2','es-ar') 'promedio gastos'

from facturas f join clientes c on f.id\_cliente = c.id\_cliente

join detalles\_factura df on f.id\_factura = df.id\_factura

where year(fecha) = @anio

group by month(fecha), year(fecha)

end

--exec pa\_clientes\_mes 2021

--3) Se quiere saber la cantidad de facturas, la facturacion total, la fecha de la primer y ultima factura por empleado y cliente, para las facturas de este año que oscilen entre los codigos 1 y 10

--ordenado por vendedor, cantidad de ventas en forma descendente y cliente.

select e.nombre+' '+e.apellido 'empleado', c.nombre+' '+c.apellido 'cliente',

count (f.id\_factura) 'cantidad',

format(sum(cantidad\*precio\_unitario),'c2','es-ar') 'total' ,

format(min(f.fecha),'dd/mm/yyyy') 'primera',

format(max(f.fecha),'dd/mm/yyyy') 'ultima'

from empleados e join facturas f on e.id\_empleado=f.id\_empleado

join detalles\_factura df on df.id\_factura=f.id\_factura

join clientes c on c.id\_cliente=f.id\_cliente

where year(fecha)=year(getdate()) and f.id\_factura between 1 and 10

group by e.nombre+' '+e.apellido, c.nombre+' '+c.apellido

order by 1, 3 desc, 2

--4) Verificar si el cliente tiene obra social, si tiene mostrar los datos de la obra social

create proc pa\_tiene\_os

@codigo int = 1

as

begin

declare @tiene int = 0

select @tiene = 1

from clientes c join planes p on c.id\_cliente = p.id\_plan

where c.id\_cliente = @codigo and

c.id\_plan is not null

if @tiene = 1

begin

print 'tiene obra social'

select o.nombre 'obra social', p.nombre 'nombre del plan', o.id\_obra\_social 'codigo os', p.id\_plan 'codigo plan', trim(str(d.descuento\*100) + '% en ' + ts.tipo) as 'descuentos en', l.localidad + ', ' + pr.provincia as 'lugar del descuento','desde: ' + convert(varchar,format(fecha\_desde,'dd/mm/yyyy'),131) + ' | hasta: ' + convert(varchar,format(fecha\_hasta,'dd/mm/yyyy'),131) as 'fechas -> desde | hasta'

from clientes c join planes p on c.id\_plan = p.id\_plan

join obras\_sociales o on p.id\_obra\_social = o.id\_obra\_social

join descuentos d on p.id\_plan = d.id\_plan

join localidades l on d.cod\_localidad = l.cod\_localidad

join provincias pr on l.cod\_provincia = pr.cod\_provincia

join tipos\_suministro ts on d.id\_tipo\_suministro = ts.id\_tipo\_suministro

where c.id\_cliente = @codigo

end

else

begin

select 'no tiene obra social'

end

end

--exec pa\_tiene\_os 1

--5) Se quiere saber el precio promedio de medicamento, el total recaudado en medicamentos de venta libre, por obra social en los que lo recaudado fue superior a lo recaudado en medicamentos que no sean de venta libre

create proc pa\_med\_os

as

begin

select ts.tipo 'medicamento', str(pl.id\_obra\_social) + ' ' + os.nombre 'obra social',

format(avg (s.precio),'c2','es-ar') 'precio promedio',

format(sum (df.precio\_unitario\*df.cantidad\*(1-df.descuento)),'c2','es-ar') 'importe total'

from suministros s join detalles\_factura df on s.codigo\_barra = df.codigo\_barra

join tipos\_suministro ts on s.id\_tipo\_suministro = ts.id\_tipo\_suministro

join descuentos d on ts.id\_tipo\_suministro = d.id\_tipo\_suministro

join planes pl on d.id\_plan = pl.id\_plan

join obras\_sociales os on pl.id\_obra\_social = os.id\_obra\_social

where s.venta\_libre = 1

group by pl.id\_obra\_social, ts.tipo, os.nombre

having sum (df.precio\_unitario\*df.cantidad\*(1-df.descuento)) >

(select sum(det.precio\_unitario\*cantidad\*(1-det.descuento))

from suministros su

join detalles\_factura det on su.codigo\_barra = det.codigo\_barra

where su.venta\_libre = 0

group by su.codigo\_barra)

or

(select sum(det.precio\_unitario\*cantidad\*(1-det.descuento))

from suministros su

join detalles\_factura det on su.codigo\_barra = det.codigo\_barra

where su.venta\_libre = 0

group by su.codigo\_barra) is null

end

--exec pa\_med\_os

--6) Emitir un listado con los datos de los médicos que no registran ninguna receta desde un año en particular que se ingresará por parámetro.

create proc pa\_medico\_año

@año int

as

select nombre + space(2) + apellido 'nombre médico', matricula 'matricula'

from medicos

where id\_medico not in (select re.id\_medico

from recetas re

where year(fecha) >= @año)

-- exec pa\_medico\_año @año = 2019

--7) Cantidades de afiliados por obra social que realizaron compras, y total de descuentos aplicados filtrando aquellas obras sociales que tengan como mínimo la cantidad de afiliados indicado por parámetro

create proc pa\_clie\_osocial

@min\_cant\_afiliados int=0

as

begin

select o.nombre 'obra social', count(c.id\_cliente) 'cantidad de afiliados',

format(sum(df.descuento\*precio\_unitario\*cantidad),'c2','es-ar')'descuento total aplicado from clientes c join planes p on c.id\_plan = p.id\_plan

join obras\_sociales o on p.id\_obra\_social = o.id\_obra\_social

join facturas f on f.id\_cliente=c.id\_cliente

join detalles\_factura df on df.id\_factura=f.id\_factura

group by o.nombre

having count(c.id\_cliente)>=@min\_cant\_afiliados

end

--exec pa\_clie\_osocial 1

--8) Listado que muestre el monto máximo, mínimo y total que gasto cada cliente el mes pasado, pero solo donde el importe total gastado sea menor a $10000

create proc pa\_mes\_pas

as

begin

select c.nombre + ' ' + c.apellido cliente,

format(max(df.precio\_unitario\*cantidad\*(1-descuento)),'c2','es-ar') 'monto maximo',

format(min(df.precio\_unitario\*cantidad\*(1-descuento)),'c2','es-ar') 'monto minimo',

format(sum(df.precio\_unitario\*cantidad\*(1-descuento)),'c2','es-ar') 'monto total'

from detalles\_factura df join facturas f on df.id\_factura = f.id\_factura

join clientes c on f.id\_cliente = c.id\_cliente

where datediff(month,fecha,getdate()) = 1

group by c.nombre + ' ' + c.apellido

having sum(df.precio\_unitario\*cantidad\*(1-descuento)) < 10000

end

--exec pa\_mes\_pas