

EVALUACIÓN	Parcial 1	GRUPO	N3F	FECHA	8/5/2023
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	- Puntos: 30 - Duración 2 horas - Sin material				

EJERCICIO 1

La heladería Popi lo contrató a Ud. Para construir la base de datos que usará su nuevo sistema de ventas. Del relevamiento, solo se obtuvo la planilla que usan actualmente y algún dato de negocio:

nroFactura	Tienda	FechaFactura	NombreTienda	idCliente	NombreCliente	CodArticulo	NomArticulo	Cant	PrecioUnitario
1	100	1/1/2024	Popi - Local Punta Carretas	99999	Juan Perez	A001	Cucurucho	3	9,99
1	200	1/1/2024	Popi - Mvd Shopping	88888	Lorena Fernandez	A002	Vasito Grand	4	4,56
2	300	1/1/2024	Popi - Tres Cruces	7777	Raul Quirk	A002	Vasito Grand	5	4,56
3	100	2/3/2024	Popi - Local Punta Carretas	99999	Juan Perez	A004	Paleta	1	3,41
4	100	2/3/2024	Popi - Local Punta Carretas	88888	Lorena Fernandez	A001	Cucurucho	3	9,99
5	100	2/3/2024	Popi - Local Punta Carretas	7777	Raul Quirk	A004	Paleta	6	3,41
4	200	5/4/2024	Popi - Tres Cruces	99999	Juan Perez	A001	Cucurucho	1	9,99
...									

- La cantidad de artículos (Cant) no puede ser nula
- Cada local empieza con la factura nro 1
- El precio unitario de los artículos es único

Se pide:

- Transforme la planilla en un MR en 3FN y construya sus tablas usando SQL DDL. Debe considerar toda restricción(constraint) que surge del MR **(5 PUNTOS)**
- Cree los índices correspondientes **(5 PUNTOS)**

EJERCICIO 2

Basado en el modelo relacional Northwind (adjunto), resolver:

#	Enunciado	Resultado Esperado					P. Max.
A	Para cada supplier, mostrar el monto promedio (quantity * unitprice) de las líneas en aquellas ordenes en que sus productos participaron de ventas. Solo aquellos Suppliers que el promedio general supere los \$400	ContactName	Prom			5	
		Yoshi Nagase	657,5137				
		Michael Björn	427,2509				
		Anne Heikkonen	425,7714				
		Chandra Leka	547,9975				
		Eliane Noz	1205.5428				
B	Mostrar el nombre de aquellos clientes que tienen ordenes en los que el monto total de la orden supera a alguno de los 5 montos máximos de descuento hecho por algún empleado en las ordenes que generó . Tip: Monto Total de una Orden: (d.Quantity*d.UnitPrice) Monto Total Descuento de una Orden: (d.Quantity*d.UnitPrice*d.Discount)	ContactName				5	
		1	Jean Fresnière				
		2	Janete Limeira				
		3	Jytte Petersen				
		4	Roland Mendel				
		5	Elizabeth Lincoln				
		6	Art Braunschweiger				
C	Mostrar aquellos empleados que tienen ordenes donde su valor de flete (Freight) es mayor a los fletes de las ordenes (de cualquier cliente) donde participaron productos que en su descripción contienen 'or'	FirstName	LastName				5
		Andrew	Fuller				
		Anne	Dodsworth				
		Janet	Leverling				
		Margaret	Peacock				
		Robert	King				
		Steven	Buchanan				
D	Mostrar los datos del empleado y los datos de su jefe, indicando la cantidad de ordenes que su jefe realizó. Si no tiene jefe, debe mostrar 0 ordenes	NomEmpleado	ApeEmpleado	NomJefe	ApeJefe	totalordenesjefe	5
		Nancy	Davolio	Andrew	Fuller	96	
		Andrew	Fuller	NULL	NULL	0	
		Janet	Leverling	Andrew	Fuller	96	
		Margaret	Peacock	Andrew	Fuller	96	
		Steven	Buchanan	Andrew	Fuller	96	
		Michael	Suyama	Steven	Buchanan	42	
		Robert	King	Steven	Buchanan	42	
		Laura	Callahan	Andrew	Fuller	96	
		Anne	Dodsworth	Steven	Buchanan	42	

SOLUCION PROPUESTA

EJERCICIO 1

```
create database Helederia
go
use Helederia
go

create table clientes(
idcliente int primary key,
nombreCliente varchar(100))

create table articulos(
codArticulo varchar(5) primary key,
nomArticulo varchar(50),
PrecioUnitario money unique)

create table tiendas(
idTienda int primary key,
NombreTienda varchar(50))

create table ventas(
nroFactura int,
idTienda int references Tiendas(idTienda),
fechaFactura date,
idCliente int references clientes(idcliente),
constraint pk_ventas primary key (nroFactura,idTienda)
)

create table detalleVentas(
nroFactura int ,
idTienda int ,
codArticulo varchar(5) references articulos(codArticulo),
cant int check(cant > 0),
preciounitario money,
constraint pk_detalle primary key(nroFactura,idTienda,codArticulo),
constraint fk_ventas foreign key(nroFactura,idTienda) references ventas(nroFactura,idtienda)
)

create index idx_ventas on Ventas(idCliente)
```

EJERCICIO 2

```
-- A Para cada supplier, mostrar el monto promedio (quantity * unitprice) de las lineas en aquellas
ordenes
--en que sus productos participaron de ventas. Solo aquellos Suppliers que el promedio supere los
$400
```

```
select s.ContactName, AVG(d.UnitPrice*d.Quantity) as Prom
from Suppliers S
inner join Products p on p.SupplierID = s.SupplierID
inner join [Order Details] d on d.ProductID = p.ProductID
```

```
group by s.ContactName  
having AVG(d.UnitPrice*d.Quantity) > 400
```

/*B) Mostrar el nombre de aquellos cliente cuyo monto total (d.Quantity*d.UnitPrice) de todas las ordenes supera a alguno de los 5 montos máximos de descuento(d.Quantity*d.UnitPrice*d.Discount) hecho por un empleado en las ordenes que generó.
*/

```
select c.ContactName  
from Customers c  
inner join Orders o on o.CustomerID = c.CustomerID  
inner join [Order Details] d on d.OrderID = o.OrderID  
group by c.ContactName  
having sum(d.Quantity*d.UnitPrice) >= ANY (  
  
select top 5  
  
sum(d.Quantity*d.UnitPrice*d.Discount) as totaldesc  
  
from Orders o  
inner join [Order Details] d on  
  
group by o.EmployeeID  
order by totaldesc desc)
```

--C) Mostrar aquellos empleados que tienen ordenes que su valor de flete (Freight) es mayor a --los fletes de las ordenes donde participaron productos que en su descripcion contienen 'or'

```
select distinct e.FirstName,e.LastName  
from Orders o  
inner join Employees e on e.EmployeeID = o.EmployeeID  
where o.Freight >= ALL(select o.Freight  
  
from Orders o  
inner join [Order Details] d on d.OrderID = o.OrderID  
inner join Products p on p.ProductID = d.ProductID  
where p.ProductName like '%or%')
```

--D
-- Mostrar los datos del empleado y los datos de su jefe, indicando la cantidad de ordenes que su jefe realizó.

-- Si no tiene jefe, debe mostrar 0 ordenes

```
select e.FirstName NomEmpleado,e.LastName ApeEmpleado,  
       j.FirstName NomJefe,j.LastName ApeJefe,  
       (select count(*)  
        from Orders o  
        where o.EmployeeID = j.EmployeeID) as totalordenesjefe  
from Employees e  
left join Employees j on e.ReportsTo = j.EmployeeID
```