

EVALUACIÓN	Parcial 1	GRUPO	N3D	FECHA	Mayo 2022
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	<ul style="list-style-type: none">- Puntos: <u>Máximo</u>: 30.- Sin Material- Duración: 2 horas				

La empresa CaBaMa es una empresa distribuidora dedicada a comercializar a clientes los productos que son fabricados por otras empresas.

A continuación se presenta la estructura de la base de datos utilizada en la empresa:

Proveedores (proveedorId, proveedorNom, proveedorCiu, proveedorDir, proveedorMail)

En esta tabla se almacena información de los proveedores. Los mismos están identificados por proveedorId . Se conoce nombre, ciudad, dirección y mail (el cual es único).

Clientes(clienteId, clienteNom, clienteCiu, clienteDir, clienteSaldo)

En esta tabla se registra información de los clientes de la empresa. Cada cliente tiene un id autonumérico, se conoce su nombre, ciudad (que no puede tener más de 30 caracteres), dirección y el saldo en dólares que tiene adeudado con la distribuidora. El saldo máximo que un cliente puede tener es de 500 dólares.

Empleados (empleadId, empleadoNom, empleadoCiu, empleadoDir, EmpleadoFechaNac)

Los empleados tienen un id autonumérico, nombre, ciudad, dirección, y fecha de nacimiento (debe ser mayor a 18 años).

Productos (productId, productoDescrip, productoStock, productoPrecio, proveedorId)

Por cada producto se cuenta con un identificador, una descripción que no puede ser vacía, un stock, precio y se conoce quién es el proveedor del producto. El stock de un producto nunca puede ser menor a 5.

Pedidos (pedidoId, clienteId, empleadId, pedidoFecha, pedidoEntrega, pedidoCiudad, costoEnvio)

Para cada pedido se maneja un id, el cliente que realizó el pedido, el empleado que lo tomó, la fecha en que se realizó el pedido y la fecha en que se entregó. Se conoce además la ciudad del pedido y el costo del mismo.

Detalle(pedId, productId, detCantidad, detPrecio, detDescuento)

Cada pedido, se relaciona con una o varias filas en la tabla detalle, la cual cuenta con un id como identificador, una referencia a producto, una cantidad para dicho producto en la fila (debe ser mayor a 0), un precio total de la línea de detalle (mayor a 1) y un descuento aplicado a la fila del detalle.

SE PIDE:

1. Script de creación de todas las tablas teniendo en cuenta las restricciones de integridad marcadas para cada una de ellas. (10 puntos)

```
create database parcialbd2_20221_n3d;
```

```
create table Proveedores (  
    proveedorId int not null primary key,  
    proveedorNom varchar(40),  
    proveedorCiu varchar(50),  
    proveedorDir varchar(40),  
    proveedorMail varchar(40) unique  
  
)
```

```
create table Clientes(  
    clienteId int identity primary key,  
    clienteNom varchar(40),  
    clienteCiu varchar(30),  
    clienteDir varchar(100),  
    clienteSaldo int check (clienteSaldo > 0 and clienteSaldo < 500)  
  
)
```

```
create table Empleados (  
    empleadId int identity primary key,  
    empleadoNom varchar(40),  
    empleadoCiu varchar(40), empleadoDir varchar(40), empleadoMail varchar(40),  
    EmpleadoFechaNac date check (DATEDIFF(YEAR, EmpleadoFechaNac, GETDATE()) > 18)  
  
)
```

```
create table Productos (  
    productId int identity primary key,  
    productoDescrip varchar(40) not null,  
    productoStock int check (productostock > 5),  
  
)
```

```
productoPrecio int ,  
provedorId int,
```

```
constraint FK_Prov_prod foreign key (provedorId) references Proveedores(provedorId)
```

```
)
```

```
create table Pedidos (  
    pedidoid int,  
    clienteId int ,  
    empleadoid int foreign key(empleadoid) references Empleados(Empleadoid),  
    pedidoFecha date,  
    pedidoEntrega date,  
    pedidoCiudad varchar(50),  
    pedidoEnvio int,  
  
    primary key (pedidoid, clienteid, empleadoid),  
)
```

```
create table Detalle(  
    pedidoid int primary key ,  
    productoid int foreign key (productoid) references Productos (productoid),  
    detCantidad int check (detCantidad > 0),  
    detPrecio int check(detprecio > 1),  
    detDescuento int,  
  
    constraint fk_pedidos_pedid foreign key (pedidoid) references Pedidos (pedidoid)  
)
```

2. Creación de índices sobre las claves foráneas que considere necesario según el criterio establecido en el curso PARA TODO EL MODELO. (2 puntos)

```
create index I1 on Productos(provedorId);  
create index I2 on Pedidos (Empleadoid);  
create index I3 on Pedidos (ClienteId);  
create index I4 on Detalle(pedidoid);  
create index I5 on Detalle (productoid)
```

-
3. Escribir el SQL necesario para ingresar un cliente en la tabla Clientes, asumiendo que la base de datos está vacía. (1 punto).

```
insert into Clientes values ('carmen', 'mvdeo', 'dir 1234', 'c@c.com', 900);
```

4. En la tabla Productos, agrandar el campo ProductoDescrip y llevarlo a 200 caracteres y que sea único. (2 puntos)

5. Resolver las siguientes consultas SQL

- a. Mostrar para cada cliente la cantidad de pedidos realizados y el total de los mismos. (4 puntos)

```
select c.clientId, c.clienteNom, count(distinct p.pedidoId) as CantidadPedidos, sum  
(d.detCantidad*d.detPrecio)  
from Clientes c, Pedidos p, Detalle d  
where p.clientId = c.clientId and d.pedidoId = p.pedidoId  
group by c.clientId, c.clienteNom
```

- b. Mostrar los datos de los productos cuyo mayor descuento aplicado supera los U\$S 30 (4 puntos)

```
SELECT productos.productDescrip,MAX(detalle.detDescuento) AS mayor_descuento  
FROM productos,detalle  
WHERE productos.productId=detalle.productId  
GROUP BY productos.productDescrip  
HAVING MAX(detalle.detDescuento) > 30
```

- c. Mostrar toda la información de los clientes que compraron algún producto en el año 2020 y no compraron ninguno en el año actual. (4 puntos)

```
select *  
from Clientes c, productos p, pedidos pe, detalle d  
where d.pedidoId = pe.pedidoId and c.clientId = pe.clientId and p.productId =  
d.productId and year(pe.pedidoEntrega) = 2020  
and c.clientId not in (select c1.clientId from Clientes c1, Pedidos p1, detalle d1  
where c1.clientId = p1.clientId and  
p1.pedidoId = d1.pedidoId and year(pe.pedidoEntrega) = GETDATE())
```

-
- d. Obtener todos los datos de los empleados que prepararon los pedidos en la fecha más antigua. (3 p)

SELECT empleados.*

FROM empleados

WHERE empleados.emplId IN (SELECT pedidos.emplId

FROM pedidos

WHERE pedidos.pedFecha=(SELECT

MIN(pedidos.pedFecha) FROM pedidos))