



Tecnicatura Universitaria
en Programación

LABORATORIO DE COMPUTACIÓN II

Trabajo Práctico Integrador

1W3 - Grupo N°6

Fecha de entrega: 24/10/2021

113227 - Ros, Natalia

112791 - Ulloa, Mariano

112710 - Uriza, Valentina

112809 - Urquiza, Juan

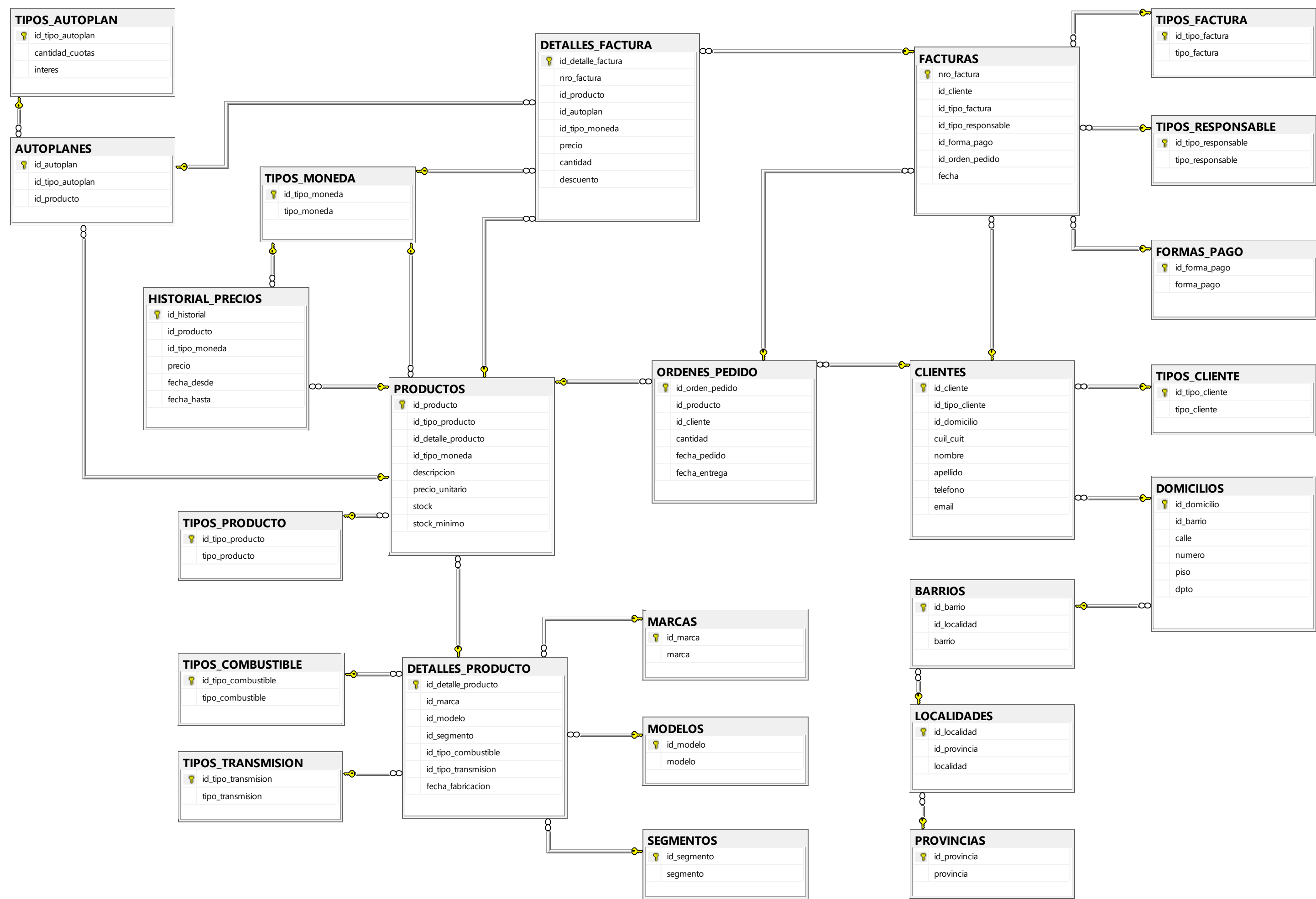
112728 - Verdugo, Carlos



Índice

Diagrama del Modelo Relacional de la Base de Datos	2
Reportes	3

Diagrama del Modelo Relacional de la Base de Datos



Reportes

--1) Crear una vista que permita ver los vehiculos y las autopartes.

```
CREATE VIEW V_VEHICULOS
AS
SELECT p.descripcion 'Vehiculo', p.precio_unitario 'Precio', p.stock 'Stock',
m.marca 'Marca', mo.modelo 'Modelo', s.segmento 'Segmento', tc.tipo_combustible 'Tipo combustible',
tt.tipo_transmision 'Tipo transmision'

FROM PRODUCTOS p, DETALLES_PRODUCTO dp, MARCAS m, MODELOS mo, SEGMENTOS s,
TIPOS_COMBUSTIBLE tc, TIPOS_TRANSMISION tt

WHERE p.id_detalle_producto = dp.id_detalle_producto AND dp.id_marca = m.id_marca
AND dp.id_modelo = mo.id_modelo AND dp.id_segmento = s.id_segmento AND dp.id_combustible =
tc.id_combustible
AND dp.id_tipo_transmision = tt.id_tipo_transmision

CREATE VIEW VIS_AUTOPARTES
AS
SELECT p.descripcion 'Autopartes', p.precio_unitario 'Precio', p.stock 'Stock',
m.marca 'Marca', mo.modelo 'Modelo'

FROM PRODUCTOS p, DETALLES_PRODUCTO dp, MARCAS m, MODELOS mo

WHERE p.id_detalle_producto = dp.id_detalle_producto AND dp.id_marca = m.id_marca
AND dp.id_modelo = mo.id_modelo
```

--2) Crear un SP que, al cargar una factura en caso de no contar con stock, generar la orden de pedido. En caso de que no se pueda efectuar ninguna de las operaciones informar a modo de error.

```
CREATE PROC SP_Generar_Pedido
@id_producto int,
@cantidad int,
@id_cliente int,
@id_tipo_factura int,
@id_tipo_responsable int,
@descuento int,
@precio money,
@id_forma_pago int
AS
BEGIN
declare @stock int
declare @nro_factura int
select @stock=stock from PRODUCTOS
SET @nro_factura = (select ISNULL(MAX(nro_factura), 0)+1 FROM FACTURAS)

begin try
if (@stock >= @cantidad)
begin
insert into FACTURAS (id_tipo_factura, id_cliente, id_tipo_responsable, id_forma_pago, fecha)
VALUES (@id_tipo_factura, @id_cliente, @id_tipo_responsable, @id_forma_pago, GETDATE())
insert INTO DETALLES_FACTURA (nro_factura, id_producto, precio, cantidad, descuento)
VALUES (@nro_factura, @id_producto, @precio, @cantidad, @descuento)
end
else
insert into ORDENES_PEDIDO (id_producto, id_cliente, cantidad, fecha_pedido, fecha_entrega)
VALUES (@id_producto, @id_cliente, @cantidad, GETDATE(), DATEADD(WEEK,3,GETDATE()))

end try
begin catch
select @@ERROR
select ERROR_MESSAGE() 'Mensaje', ERROR_SEVERITY() 'Gravedad del error', ERROR_NUMBER() 'Nº del
error'
end catch
END
```

--3) Crear un SP que muestre los totales de facturación y promedio de facturación periódicos (mensual, trimestral, semestral o anual)

```
CREATE PROC SP_TOTALES_FACTURACION
@periodo varchar(10)
AS
BEGIN
if @periodo = 'mensual'
begin
SELECT sum(precio*cantidad) 'Total Facturado',
SUM(precio*cantidad) / COUNT(DISTINCT df.nro_factura) 'Promedio facturación'
from facturas f, detalles_factura df
where f.nro_factura = df.nro_factura
AND MONTH(f.fecha) = MONTH(GETDATE())
end
if @periodo = 'trimestral'
begin
select sum(precio*cantidad) 'Total Facturado',
SUM(precio*cantidad) / COUNT(DISTINCT df.nro_factura) 'Promedio facturación'
from facturas f, detalles_factura df
where f.nro_factura=df.nro_factura
AND MONTH(f.fecha) BETWEEN MONTH(GETDATE()) AND DATEADD(MONTH, -3, GETDATE())
end
if @periodo = 'semestral'
begin
select sum(precio*cantidad) 'Total Facturado',
SUM(precio*cantidad) / COUNT(DISTINCT df.nro_factura) 'Promedio facturación'
from facturas f, detalles_factura df
where f.nro_factura=df.nro_factura
AND MONTH(f.fecha) BETWEEN MONTH(GETDATE()) AND DATEADD(MONTH, -6, GETDATE())
end
if @periodo = 'anual'
begin
select sum(precio*cantidad) 'Total Facturado',
SUM(precio*cantidad) / COUNT(DISTINCT df.nro_factura) 'Promedio facturación'
from facturas f, detalles_factura df
where f.nro_factura=df.nro_factura
AND YEAR(f.fecha) BETWEEN YEAR(GETDATE()) AND DATEADD(YEAR, -1, GETDATE())
end
END
```

--4) Crear un SP que muestre las ordenes de pedido que vencen la semana proxima. Detallar codigo de producto, tipo, descripcion, cantidad, y los datos de contacto del cliente.

```
CREATE PROCEDURE SP_CONSULTA_ORDENES_PEDIDO
AS
BEGIN
SELECT op.id_orden_pedido 'Orden de Pedido',
p.id_producto 'Codigo de Producto',
p.id_tipo_producto 'Tipo',
p.descripcion 'Descripcion',
op.cantidad 'Cantidad',
c.nombre 'Cliente',
c.telefono 'Telefono',
c.email 'E-mail'
FROM ORDENES_PEDIDO op
JOIN PRODUCTOS p ON op.id_producto = p.id_producto
JOIN CLIENTES c ON op.id_cliente = c.id_cliente
WHERE fecha_entrega BETWEEN GETDATE() AND DATEADD(WEEK, 1, GETDATE())
END
```

--5) Crear un trigger que muestre el stock disponible de un artículo luego de efectuar la venta del mismo

```
create trigger TR_VENTA_STOCK
on detalles_factura for insert
as
declare @cod_prod int, @cantidad int, @stock int
select @cod_prod = p.id_producto, @cantidad = cantidad , @stock = stock
from inserted i
join productos p on p.id_producto = i.id_producto
begin
update PRODUCTOS set stock = @stock-@cantidad where id_producto = @cod_prod
end
```