Icono

Descripción generada automáticamente

UNIVERSIDAD – TECNOLÓGICA – NACIONAL

FACULTAD – REGIONAL – CÓRDOBA

TECNICATURA – UNIVERSITARIA - EN – PROGRAMACIÓN

LABORATORIO DE COMPUTACIÓN II

TRABAJO PRÁCTICO I - PARTE 2

FECHA DE ENTREGA:   
 6 de Octubre del 2022 a las 23:55hs

COMISIÓN 1W3 (TURNO NOCHE)

INTEGRANTES GRUPO N° 1:

113842 - Ahumada Federico

113196 - Guizzo Diego Nicolás

114098 - Villarruel Matías Nicolás

114266 - Vitelli Conrado

CONTENIDO

[ENUNCIADO 2](#_Toc116125166)

[DIAGRAMA DE BASE DE DATOS 3](#_Toc116125167)

[CONSULTAS 4](#_Toc116125168)

[Ahumada Federico 4](#_Toc116125169)

[Guizzo Diego Nicolás 6](#_Toc116125170)

[Villarruel Matías Nicolás 7](#_Toc116125171)

[Vitelli Conrado 8](#_Toc116125172)

[SCRIPT + INSERTS 17](#_Toc116125173)

# ENUNCIADO

Sistema de Venta y Control de Stock de una Automotriz

Una fábrica automotriz necesita un sistema para el control del stock y de lo que vende, como así también de sus clientes, que pueden ser consumidor final, una empresa, un concesionario, o una vendedora de autopartes.

Los productos que la misma fabrica y vende son vehículos terminados o autopartes. De las autopartes se cuenta con un stock y se conoce el mínimo de stock que debe haber en inventario, en cambio para vehículos no se cuenta con stock previo.

Cuando un cliente quiere comprar vehículos, o autopartes que no se encuentran en stock se genera una orden de pedido, y se le informa al cliente la fecha de entrega. Para el caso de autopartes en donde la cantidad que haya en inventario cubra la cantidad pedida por el cliente, solo se genera una orden de pedido en el caso de que la nueva cantidad de inventario sea menor al stock mínimo.

Al momento de realizar la factura se asocia la orden de pedido en caso de que haya alguna.

Los clientes que son consumidores finales tienen la posibilidad de entrar en un autoplan, lo cual implica que al precio del producto se le aplicará una tasa de interés dependiendo de la cantidad de cuotas del plan.

Por otro lado, las vendedoras de autopartes reciben descuentos en los productos por compras realizadas al por mayor.

La empresa además necesita conocer el historial de precios de sus productos.

# DIAGRAMA DE BASE DE DATOS

# CONSULTAS

## Ahumada Federico

1. Se necesita saber que impacto tuvieron las ventas a concesionarias en articulos mayores

o iguales a $10000 en la provincia de Córdoba en septiembre.

-Detallar el promedio, la cantidad de ventas y el monto facturado por empleado.

CONDICION: Mostrar los datos solicitados si el promedio de ventas supera al

promedio de venta del mes anterior.

CREATE VIEW VIS\_IMPACTO\_VTAS\_SEPTIEMBRE\_POR\_EMPLEADO

AS

select e2.empleado'EMPLEADO', year(f.fecha\_factura)'AÑO', month(f.fecha\_factura)'MES', prov.provincia'PROVINCIA', convert(int,avg(d.cantidad\*d.precio\_unitario))'PROMEDIO FACTURACION', sum(d.cantidad\*d.precio\_unitario)'FACTURACION', count(f.id\_factura)'CANT. VENTAS'

from pedidos p

join facturas f on p.id\_factura = f.id\_factura

join detalles d on p.id\_pedido = d.id\_pedido

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

join personas per on per.id\_persona = c.id\_persona

join barrios b on b.id\_barrio = per.id\_barrio

join ciudades ciu on ciu.id\_ciudad = b.id\_ciudad

join provincias prov on prov.id\_provincia = ciu.id\_provincia

join

(select c.id\_cliente

from clientes c

join tipo\_clientes tc on tc.id\_tipo\_cliente = c.id\_tipo\_cliente

where tc.tipo\_cliente = 'concesionario'

) as c2 on c2.id\_cliente = c.id\_cliente

join

(select e.id\_empleado,upper(substring(p.nombre,1,1))+'.'+

upper(substring(p.apellido,1,1))+lower(substring(p.apellido,2,100))'empleado'

from empleados e

join personas p on p.id\_persona = e.id\_persona

join tipo\_empleados te on te.id\_tipo\_empleado = e.id\_tipo\_empleado

where te.tipo\_empleado = 'empleado'

and id\_empleado in (select id\_empleado

from pedidos

where id\_factura is not null

)

) as e2 on e2.id\_empleado = p.id\_empleado

where datediff(month,f.fecha\_factura,getdate())<2 and d.id\_factura is not null

and prov.provincia = 'Córdoba'

group by e2.empleado, year(f.fecha\_factura), month(f.fecha\_factura),prov.provincia

having avg(d.cantidad\*d.precio\_unitario)>=

(select avg(d.cantidad\*d.precio\_unitario)

from facturas f

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

where datediff(month,f.fecha\_factura,getdate())=2

)

-- EJECUCION DE LA VISTA

select \* from dbo.VIS\_IMPACTO\_VTAS\_SEPTIEMBRE\_POR\_EMPLEADO

1. Buscando implementar el email marketing, se requiere saber el listado

de clientes con e-mail registrado en la base de datos que hayan hecho órdenes de compra

en los últimos 30 días. El equipo se contactará con ellos a fin de ofrecer cupones de

descuento en productos seleccionados.

Se aplicarán descuentos del 7% a consumidor final y del 13% a vendedores de autopartes.

CREATE VIEW VIS\_ENCRYPT\_EMAIL\_MARKETING

(tipo,cliente,provincia,ciudad,barrio,contacto,descuento)

WITH ENCRYPTION

AS

select c2.tcliente TIPO, upper(substring(p.nombre,1,1))+lower(substring(p.apellido,2,100))+' '+

upper(substring(p.apellido,1,1))+lower(substring(p.apellido,2,100))CLIENTE,

prov.provincia PROVINCIA, ciu.ciudad CIUDAD, upper(substring(b.barrio,1,1))+

lower(substring(b.barrio,2,100)) BARRIO, lower(tc.contacto) CONTACTO,'17%' DESCUENTO

from personas p

join clientes c on p.id\_persona = c.id\_persona

join tipo\_clientes tc on tc.id\_tipo\_cliente = c.id\_tipo\_cliente

join barrios b on b.id\_barrio = p.id\_barrio

join ciudades ciu on ciu.id\_ciudad = b.id\_ciudad

join provincias prov on prov.id\_provincia = ciu.id\_provincia

join

(select c.id\_cliente, tc.tipo\_cliente 'tcliente'

from clientes c

join tipo\_clientes tc on tc.id\_tipo\_cliente = c.id\_tipo\_cliente

where tc.tipo\_cliente = 'consumidor final'

) as c2 on c2.id\_cliente = c.id\_cliente

where c.id\_cliente in (select id\_cliente

from pedidos pe

join facturas f on pe.id\_factura = f.id\_factura

where datediff(day,f.fecha\_factura,getdate())<=30

)

and tc.contacto LIKE '%@%' and tc.contacto is not null

union

select c2.tcliente, upper(substring(p.nombre,1,1))+lower(substring(p.apellido,2,100))+' '+

upper(substring(p.apellido,1,1))+lower(substring(p.apellido,2,100)),

prov.provincia, ciu.ciudad, upper(substring(b.barrio,1,1))+

lower(substring(b.barrio,2,100)), lower(tc.contacto),'13%'

from personas p

join clientes c on p.id\_persona = c.id\_persona

join tipo\_clientes tc on tc.id\_tipo\_cliente = c.id\_tipo\_cliente

join barrios b on b.id\_barrio = p.id\_barrio

join ciudades ciu on ciu.id\_ciudad = b.id\_ciudad

join provincias prov on prov.id\_provincia = ciu.id\_provincia

join

(select c.id\_cliente, tc2.tipo\_cliente 'tcliente'

from clientes c

join tipo\_clientes tc2 on tc2.id\_tipo\_cliente = c.id\_tipo\_cliente

where tc2.tipo\_cliente = 'autopartes'

) as c2 on c2.id\_cliente = c.id\_cliente

where c.id\_cliente in (select id\_cliente

from pedidos pe

join facturas f on pe.id\_factura = f.id\_factura

where datediff(day,f.fecha\_factura,getdate())<=30

)

and tc.contacto LIKE '%@%' and tc.contacto is not null

-- EJECUCION DE LA VISTA

select \* from dbo.VIS\_ENCRYPT\_EMAIL\_MARKETING

## Guizzo Diego Nicolás

-- A los efcetos de evaluar el objetivo de ventas por volumen e importe y determniación de Bono para empleados anual en funcion de su histórico de pedidos, se necesita determinar una consulta que verifique

-- aquellos empleados cuyo promedio de pedidos sea mayor al promedio de pedidos general de este año.

select year(fecha\_pedido) as 'Año',pf.apellido 'Apellido',sum(cantidad\*precio\_unitario) as 'Importe total'

from PEDIDOS P join DETALLES dp on p.id\_pedido=dp.id\_pedido

join empleados e on p.id\_empleado=e.id\_empleado

join PERSONAS pf on e.id\_empleado = pf.id\_persona

group by year(fecha\_pedido),pf.apellido

having sum(cantidad\*precio\_unitario)/count(distinct p.id\_pedido)<

(select sum(cantidad\*precio\_unitario)/count(distinct p.id\_pedido)

from PEDIDOS p1

join DETALLES dp1 on p1.id\_pedido = dp1.id\_pedido

where year(p1.fecha\_pedido) = year(p.fecha\_pedido)

)

-- La gente de Finanzas necesita llevar un control de las distintas formas de cobro de las ventas realizadas para evaluar posibles inversiones de efectivo

-- a lo largo del año, para ello require una vista que le permita obtener la información:

CREATE VIEW vista\_inv\_efectivo

as

select format(p.fecha\_pedido,'dd/MM/yyyy')'FECHA'

, dp.id\_pedido 'NRO\_PEDIDO',m.id\_metodo\_pago 'CODIGO\_METODO\_PAGO'

,m.metodo\_pago 'METODO\_DE\_PAGO'

,sum(dp.precio\_unitario \* dp.cantidad) 'IMPORTE'

from

detalles dp

join PEDIDOS p on dp.id\_pedido= p.id\_pedido

join FACTURAS f on p.id\_factura= f.id\_factura

join METODO\_PAGOS m on f.id\_metodo\_pago = m.id\_metodo\_pago

where year(p.fecha\_pedido) = YEAR(GETDATE())

group by m.id\_metodo\_pago,

m.metodo\_pago,

dp.id\_pedido,

format(p.fecha\_pedido,'dd/MM/yyyy')

select \* from vista\_inv\_efectivo

## Villarruel Matías Nicolás

--montos totales mensuales

--facturados por cliente y además del

--promedio de ese monto y el promedio de precio

--de artículos Todos esto

--datos correspondientes a período que va desde

--el 1° de febrero al 30 de

--agosto de cada año. Sólo muestre los datos si

--esos montos totales son

--superiores o iguales al promedio global.

set dateformat DMY

select c.id\_cliente as 'id cliente', pers.apellido +', '+pers.nombre as 'Cliente',

SUM(d.cantidad\*d.precio\_unitario) as 'Total mensual',

AVG(d.cantidad\*d.precio\_unitario) as 'promedio mensual',

AVG(d.precio\_unitario) as 'Precio promedio por producto',

MONTH(f.fecha\_factura) as 'Mes'

from DETALLES d inner join PRODUCTOS p on d.id\_producto=p.id\_producto

inner join FACTURAS f on f.id\_factura=d.id\_detalle inner join PEDIDOS ped on ped.id\_factura = f.id\_factura

inner join CLIENTES c on c.id\_cliente=ped.id\_cliente inner join PERSONAS pers on

pers.id\_persona = c.id\_persona

where DAY(f.fecha\_factura) between 1 and 30 and MONTH(f.fecha\_factura) between 1 and 8

group by c.id\_cliente, pers.apellido,pers.nombre,MONTH(f.fecha\_factura)

having SUM(d.cantidad\*d.precio\_unitario)>= AVG(d.cantidad\*d.precio\_unitario)

--detalle facturas y vendedores

-- muestre cuánto fue el

--total anual facturado por

--cada vendedor, para los casos en que el nombre

--de vendedor no comience

--con letras que van de la ‘B’ a la ‘M’, que los id de facturas

--oscilen entre 5 y 25 y que

--el promedio del monto facturado sea inferior al

--promedio de ese año.

select e.id\_empleado as 'id vendedor', pers.apellido +', '+pers.nombre as 'Vendedor',

SUM(d.cantidad\*d.precio\_unitario) as 'Total Anual',

YEAR(f.fecha\_factura) as 'Año'

from DETALLES d inner join PRODUCTOS p on d.id\_producto=p.id\_producto

inner join FACTURAS f on f.id\_factura=d.id\_detalle inner join PEDIDOS ped on ped.id\_factura = f.id\_factura

inner join EMPLEADOS e on e.id\_empleado=ped.id\_empleado inner join PERSONAS pers on

pers.id\_persona = e.id\_persona

where pers.nombre not like '[B-M]%' and f.id\_factura between 5 and 25

group by e.id\_empleado, pers.apellido,pers.nombre, YEAR(f.fecha\_factura)

having AVG(d.precio\_unitario\*d.cantidad)<(select AVG(d.precio\_unitario\*d.cantidad)

from DETALLES d

inner join FACTURAS f on d.id\_factura= f.id\_factura

)

## Vitelli Conrado

--Mostrar el promedio de facturacion por vendedor en lo que va del mes junto con el nombre y el ID --del empleado que les corresponda, Siempre y cuando:--1) El apellido del empleado empiece con letras entre "A" y "L"--2) El promedio debe ser mayor al promedio de facturacion por vendedor del mes anteriorSELECT e.id\_empleado as 'ID EMPLEADO', pers.apellido + ', ' + pers.nombre as 'NOMBRE EMPLEADO', AVG(d.cantidad\*d.precio\_unitario) as 'PROM. FACTURADO ESTE MES'FROM DETALLES d join PRODUCTOS p on d.id\_producto=p.id\_producto

join FACTURAS f on f.id\_factura=d.id\_detalle join PEDIDOS ped on ped.id\_factura = f.id\_factura

join EMPLEADOS e on e.id\_empleado=ped.id\_empleado join PERSONAS pers on

pers.id\_persona = e.id\_personaWHERE pers.apellido like '[A-L]%' and YEAR(f.id\_factura) = YEAR(GETDATE()) and MONTH(f.id\_factura) = MONTH(GETDATE())GROUP BY e.id\_empleado, pers.apellido + ', ' + pers.nombreHAVING AVG(d.cantidad\*d.precio\_unitario) > ( SELECT AVG(d.cantidad\*d.precio\_unitario)

FROM DETALLES d

WHERE YEAR(d.id\_factura) = YEAR(GETDATE()) and

MONTH(d.id\_factura) = MONTH(GETDATE())-1

)

--Listar en una misma tabla de resultados:

--valor promedio de todos los productos filtrados por su descripcion

--(Solo de los prodcutos que valen mas de $1300, que estén en stock y sean modelos de los ultimo 10 años.)

--Valor total de todos los productos filtrados por su descripcion, que estén en stock

--y sean modelos de los ultimo 10 años

--De cada uno de los productos, el precio mínimo histórico al cual se vendieron

--De cada uno de los productos, el precio máximo histórico al cual se vendieron

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'baul') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'baul') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'baul' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'baul' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'capot') as 'PRECIO MINIMO',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'capot') as 'PRECIO MÁXIMO',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'capot' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'capot' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'carroceria') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'carroceria') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'carroceria' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'carroceria' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'freno') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'freno') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'freno' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'freno' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'luneta') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'luneta') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'luneta' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'luneta' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'motor') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'motor') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'motor' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'motor' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'puerta') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'puerta') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'puerta' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'puerta' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'parabrisa') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'parabrisa') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'parabrisa' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'parabrisa' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'rueda') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'rueda') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'rueda' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'rueda' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'ventanilla') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like 'ventanilla') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like 'ventanilla' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like 'ventanilla' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like '3 puertas') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like '3 puertas') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like '3 puertas' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like '3 puertas' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

UNION

SELECT p.descripcion as 'DESCRIPCIÓN',

(SELECT MIN(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like '5 puertas') as 'PRECIO MINIMO HIST.',

(SELECT MAX(d.precio\_unitario)

FROM PRODUCTOS p JOIN DETALLES d ON d.id\_producto=p.id\_producto

WHERE p.descripcion like '5 puertas') as 'PRECIO MÁXIMO HIST.',

SUM(p.precio\_vta) as 'VALOR TOTAL',

(SELECT AVG(precio\_vta)

FROM PRODUCTOS

WHERE descripcion like '5 puertas' and precio\_vta >= 1300 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))) as 'VALOR PROMEDIO'

FROM PRODUCTOS p

WHERE descripcion like '5 puertas' and precio\_vta >= 1000 and stock != 0

and (anio between YEAR(GETDATE())-10 and YEAR(GETDATE()))

GROUP BY descripcion

# SCRIPT + INSERTS

CREATE DATABASE AUTOMOTRIZ

GO

USE AUTOMOTRIZ;

GO

--CREACION TABLAS

CREATE TABLE PROVINCIAS(

id\_provincia int IDENTITY(1,1),

provincia varchar(100) NOT NULL,

CONSTRAINT pk\_provincias PRIMARY KEY (id\_provincia)

)

CREATE TABLE CIUDADES(

id\_ciudad int IDENTITY(1,1),

ciudad varchar(100) NOT NULL,

id\_provincia int NOT NULL,

CONSTRAINT pk\_ciudades PRIMARY KEY (id\_ciudad),

CONSTRAINT fk\_ciudades\_provincias FOREIGN KEY (id\_provincia)

REFERENCES PROVINCIAS (id\_provincia)

)

CREATE TABLE BARRIOS(

id\_barrio int IDENTITY(1,1),

barrio varchar(100) NOT NULL,

id\_ciudad int NOT NULL,

CONSTRAINT pk\_barrios PRIMARY KEY (id\_barrio),

CONSTRAINT fk\_barrios\_ciudades FOREIGN KEY (id\_ciudad)

REFERENCES CIUDADES (id\_ciudad)

)

CREATE TABLE PERSONAS(

id\_persona int IDENTITY(1,1),

nombre varchar(100) NOT NULL,

apellido varchar(100) NOT NULL,

documento bigint,

contacto varchar(200),

dire\_persona varchar(100),

id\_barrio int,

CONSTRAINT pk\_personas PRIMARY KEY (id\_persona),

CONSTRAINT fk\_personas\_barrios FOREIGN KEY (id\_barrio)

REFERENCES BARRIOS (id\_barrio)

)

CREATE TABLE TIPO\_CLIENTES(

id\_tipo\_cliente int IDENTITY(1,1),

tipo\_cliente varchar(100) NOT NULL,

razon\_social varchar(100),

dire\_empresa varchar(100),

cuit bigint,

contacto varchar(200),

descuento numeric(5,2),

id\_barrio int,

CONSTRAINT pk\_tipo\_clientes PRIMARY KEY (id\_tipo\_cliente),

CONSTRAINT fk\_tipo\_clientes\_barrios FOREIGN KEY (id\_barrio)

REFERENCES BARRIOS (id\_barrio)

)

CREATE TABLE CLIENTES(

id\_cliente int IDENTITY(1,1),

id\_persona int NOT NULL,

id\_tipo\_cliente int NOT NULL,

CONSTRAINT pk\_clientes PRIMARY KEY (id\_cliente),

CONSTRAINT fk\_clientes\_personas FOREIGN KEY (id\_persona)

REFERENCES PERSONAS (id\_persona),

CONSTRAINT fk\_clientes\_tipo\_clientes FOREIGN KEY (id\_tipo\_cliente)

REFERENCES TIPO\_CLIENTES (id\_tipo\_cliente)

)

CREATE TABLE TIPO\_EMPLEADOS(

id\_tipo\_empleado int IDENTITY(1,1),

tipo\_empleado varchar(100) NOT NULL,

CONSTRAINT pk\_tipo\_empleados PRIMARY KEY (id\_tipo\_empleado)

)

CREATE TABLE EMPLEADOS(

id\_empleado int IDENTITY(1,1),

id\_persona int NOT NULL,

id\_tipo\_empleado int NOT NULL,

cuit bigint NOT NULL,

CONSTRAINT pk\_empleados PRIMARY KEY (id\_empleado),

CONSTRAINT fk\_empleados\_tipo\_empleados FOREIGN KEY (id\_tipo\_empleado)

REFERENCES TIPO\_EMPLEADOS (id\_tipo\_empleado),

CONSTRAINT fk\_empleados\_personas FOREIGN KEY (id\_persona)

REFERENCES PERSONAS (id\_persona)

)

CREATE TABLE MARCAS(

id\_marca int IDENTITY(1,1),

marca varchar(100) NOT NULL,

CONSTRAINT pk\_marcas PRIMARY KEY (id\_marca)

)

CREATE TABLE TIPO\_PRODUCTOS(

id\_tipo\_producto int IDENTITY(1,1),

tipo\_producto varchar(100) NOT NULL,

CONSTRAINT pk\_tipo\_productos PRIMARY KEY (id\_tipo\_producto)

)

CREATE TABLE PRODUCTOS(

id\_producto int IDENTITY(1,1),

descripcion varchar(100) NOT NULL,

anio int NOT NULL,

stock bit NOT NULL,

precio\_vta numeric (10,2) NOT NULL,

id\_marca int NOT NULL,

id\_tipo\_producto int NOT NULL,

CONSTRAINT pk\_productos PRIMARY KEY (id\_producto),

CONSTRAINT fk\_productos\_marcas FOREIGN KEY (id\_marca)

REFERENCES MARCAS (id\_marca),

CONSTRAINT fk\_productos\_tipo\_productos FOREIGN KEY (id\_tipo\_producto)

REFERENCES TIPO\_PRODUCTOS (id\_tipo\_producto)

)

CREATE TABLE METODO\_PAGOS(

id\_metodo\_pago int IDENTITY(1,1),

metodo\_pago varchar(100),

CONSTRAINT pk\_metodo\_pagos PRIMARY KEY (id\_metodo\_pago)

)

CREATE TABLE TIPO\_ENVIOS(

id\_tipo\_envio int IDENTITY(1,1),

tipo\_envio varchar(100),

CONSTRAINT pk\_tipo\_envios PRIMARY KEY (id\_tipo\_envio)

)

CREATE TABLE FACTURAS(

id\_factura int IDENTITY(1,1),

fecha\_factura datetime NOT NULL,

id\_metodo\_pago int NOT NULL,

id\_tipo\_envio int,

CONSTRAINT pk\_facturas PRIMARY KEY (id\_factura),

CONSTRAINT fk\_facturas\_metodo\_pagos FOREIGN KEY (id\_metodo\_pago)

REFERENCES METODO\_PAGOS (id\_metodo\_pago),

CONSTRAINT fk\_facturas\_tipo\_envios FOREIGN KEY (id\_tipo\_envio)

REFERENCES TIPO\_ENVIOS (id\_tipo\_envio)

)

CREATE TABLE PEDIDOS(

id\_pedido int IDENTITY(1,1),

id\_factura int,

id\_empleado int NOT NULL,

id\_cliente int NOT NULL,

fecha\_pedido datetime NOT NULL,

fecha\_entrega datetime NOT NULL,

CONSTRAINT pk\_pedidos PRIMARY KEY (id\_pedido),

CONSTRAINT fk\_pedidos\_facturas FOREIGN KEY (id\_factura)

REFERENCES FACTURAS (id\_factura),

CONSTRAINT fk\_pedidos\_empleados FOREIGN KEY (id\_empleado)

REFERENCES EMPLEADOS (id\_empleado),

CONSTRAINT fk\_pedidos\_clientes FOREIGN KEY (id\_cliente)

REFERENCES CLIENTES (id\_cliente),

)

CREATE TABLE DETALLES(

id\_detalle int IDENTITY(1,1),

id\_factura int,

id\_pedido int NOT NULL,

id\_producto int NOT NULL,

precio\_unitario numeric(10,2) NOT NULL,

cantidad smallint NOT NULL,

CONSTRAINT pk\_detalles PRIMARY KEY (id\_detalle),

CONSTRAINT fk\_detalles\_facturas FOREIGN KEY (id\_factura)

REFERENCES FACTURAS (id\_factura),

CONSTRAINT fk\_detalles\_pedidos FOREIGN KEY (id\_pedido)

REFERENCES PEDIDOS (id\_pedido),

CONSTRAINT fk\_detalles\_productos FOREIGN KEY (id\_producto)

REFERENCES PRODUCTOS (id\_producto)

)

;

--Inserts

USE AUTOMOTRIZ

insert into provincias (provincia) VALUES ('Buenos Aires'),('Catamarca'),('Chaco'),('Chubut'),('Córdoba'),('Corrientes'),('Entre Ríos'),('Formosa'),('Jujuy'),('La Pampa'),('La Rioja'),('Mendoza'),('Misiones'),('Neuquén'),('Río Negro'),('Salta'),('San Juan'),('San Luis'),('Santa Cruz'),('Santa Fe'),('Santiago del Estero'),('Tierra del Fuego'),('Tucumán');

insert into ciudades (ciudad,id\_provincia) VALUES ('Achiras',5),('Agua de Oro', 5),('Alcira Gigena',5),('Almafuerte',5),('Alpa Corral',5),('Alta Gracia',5),('Amboy',5),('Anisacate',5),('Arroyito',5),('Arroyo de los Patos',5),('Ascochinga',5),('Athos Pampa',5),('Balnearia',5),('Bell Ville',5),('Bialet Masse',5),('Cabalango',5),('Calmayo',5),('Capilla del Monte',5),('Casa Grande',5),('Cerro Colorado',5),('Charbonier',5),('Colonia Caroya',5),('Copacabana',5),('Cordoba Capital',5),('Cosquin',5),('Cruz Alta',5),('Cruz Chica',5),('Cruz del Eje',5),('Cruz Grande',5),('Cuesta Blanca',5),('Dean Funes',5),('Del Campillo',5),('Despeñaderos',5),('El Durazno',5),('Embalse',5),('General Deheza',5),('General Levalle',5),('Hernando',5),('Huerta Grande',5),('Huinca Renanco',5),('Icho Cruz',5),('Intiyaco',5),('Ischilin',5),('James Craik',5),('Jesus Maria',5),('La Bolsa',5),('Laboulaye',5),('La Calera',5),('La Carlota',5),('La Cesira',5),('La Cruz',5),('La Cumbre',5),('La Cumbrecita',5),('La Falda',5),('La Granja',5),('La Paisanita',5),('La Paz Cordoba',5),('La Poblacion',5),('Las Albacas', 5),('Las Caleras',5),('Las Calles',5),('La Serranita',5),('Las Rabonas',5),('Las Tapias',5),('Loma Bola',5),('Los Cocos',5),('Los Condores',5),('Los Hornillos',5),('Los Molinos',5),('Los Pozos',5),('Los Reartes',5),('Loza Corral',5),('Marcos Juarez',5),('Mayu Sumaj',5),('Mendiolaza',5),('Mina Clavero',5),('Miramar Cordoba',5),('Molinari',5),('Monte Maiz',5),('Morteros',5),('Nono',5),('Oncativo',5),('Ongamira',5),('Panaholma',5),('Potrero de Garay',5),('Quilino',5),('Rio Ceballos',5),('Rio Cuarto',5),('Rio de los Sauces',5),('Rio Tercero',5),('Saldan',5),('Salsacate',5),('Salsipuedes',5),('San Antonio de Arredondo',5),('San Carlos Minas',5),('San Clemente',5),('San Esteban',5),('San Francisco',5),('San Javier de Traslasierra',5),('San Jose de la Dormida',5),('San Marcos Sierras',5),('San Roque',5),('Santa Cruz del Lago',5),('Santa Maria de Punilla',5),('Santa Monica',5),('Santa Rosa de Calamuchita',5),('Sinsacate',5),('Tala Huasi',5),('Tancacha',5),('Tanti',5),('Unquillo',5),('Valle Hermoso',5),('Villa Alpina',5),('Villa Allende',5),('Villa Animi',5),('Villa Ascasubi',5),('Villa Berna',5),('Villa Carlos Paz',5),('Villa Ciudad America',5),('Villa Ciudad Parque',5),('Villa Cura Brochero',5),('Villa de las Rosas',5),('Villa del Dique',5),('Villa del Soto',5),('Villa del Totoral',5),('Villa de Maria',5),('Villa Dolores',5),('Villa General Belgrano',5),('Villa Giardino',5),('Villa Los Aromos',5),('Villa Maria',5),('Villa Parque Siquiman',5),('Villa Rumipal',5),('Villa Tulumba',5),('Yacanto Calamuchita',5),('Yacanto Traslasierra',5);

insert into barrios (barrio,id\_ciudad) VALUES ('SUR',1),('NORTE',1),('ESTE',1),('OESTE',1),('SUR',2),('NORTE',2),('ESTE',2),('OESTE',2),('SUR',3),('NORTE',3),('ESTE',3),('OESTE',3),('SUR',4),('NORTE',4),('ESTE',4),('OESTE',4),('SUR',5),('NORTE',5),('ESTE',5),('OESTE',5),('SUR',6),('NORTE',6),('ESTE',6),('OESTE',6),('SUR',7),('NORTE',7),('ESTE',7),('OESTE',7),('SUR',8),('NORTE',8),('ESTE',8),('OESTE',8),('SUR',9),('NORTE',9),('ESTE',9),('OESTE',9),('SUR',10),('NORTE',10),('ESTE',10),('OESTE',10),('SUR',11),('NORTE',11),('ESTE',11),('OESTE',11),('SUR',12),('NORTE',12),('ESTE',12),('OESTE',12),('SUR',13),('NORTE',13),('ESTE',13),('OESTE',13),('SUR',14),('NORTE',14),('ESTE',14),('OESTE',14),('SUR',15),('NORTE',15),('ESTE',15),('OESTE',15),('SUR',16),('NORTE',16),('ESTE',16),('OESTE',16),('SUR',17),('NORTE',17),('ESTE',17),('OESTE',17),('SUR',18),('NORTE',18),('ESTE',18),('OESTE',18),('SUR',19),('NORTE',19),('ESTE',19),('OESTE',19),('SUR',20),('NORTE',20),('ESTE',20),('OESTE',20),('SUR',21),('NORTE',21),('ESTE',21),('OESTE',21),('SUR',22),('NORTE',22),('ESTE',22),('OESTE',22),('SUR',23),('NORTE',23),('ESTE',23),('OESTE',23),('SUR',24),('NORTE',24),('ESTE',24),('OESTE',24),('SUR',25),('NORTE',25),('ESTE',25),('OESTE',25),('SUR',26),('NORTE',26),('ESTE',26),('OESTE',26),('SUR',27),('NORTE',27),('ESTE',27),('OESTE',27),('SUR',28),('NORTE',28),('ESTE',28),('OESTE',28),('SUR',29),('NORTE',29),('ESTE',29),('OESTE',29),('SUR',30),('NORTE',30),('ESTE',30),('OESTE',30),('SUR',31),('NORTE',31),('ESTE',31),('OESTE',31),('SUR',32),('NORTE',32),('ESTE',32),('OESTE',32),('SUR',33),('NORTE',33),('ESTE',33),('OESTE',33),('SUR',34),('NORTE',34),('ESTE',34),('OESTE',34),('SUR',35),('NORTE',35),('ESTE',35),('OESTE',35),('SUR',36),('NORTE',36),('ESTE',36),('OESTE',36),('SUR',37),('NORTE',37),('ESTE',37),('OESTE',37),('SUR',38),('NORTE',38),('ESTE',38),('OESTE',38),('SUR',39),('NORTE',39),('ESTE',39),('OESTE',39),('SUR',40),('NORTE',40),('ESTE',40),('OESTE',40),('SUR',41),('NORTE',41),('ESTE',41),('OESTE',41),('SUR',42),('NORTE',42),('ESTE',42),('OESTE',42),('SUR',43),('NORTE',43),('ESTE',43),('OESTE',43),('SUR',44),('NORTE',44),('ESTE',44),('OESTE',44),('SUR',45),('NORTE',45),('ESTE',45),('OESTE',45),('SUR',46),('NORTE',46),('ESTE',46),('OESTE',46),('SUR',47),('NORTE',47),('ESTE',47),('OESTE',47),('SUR',48),('NORTE',48),('ESTE',48),('OESTE',48),('SUR',49),('NORTE',49),('ESTE',49),('OESTE',49),('SUR',50),('NORTE',50),('ESTE',50),('OESTE',50),('SUR',51),('NORTE',51),('ESTE',51),('OESTE',51),('SUR',52),('NORTE',52),('ESTE',52),('OESTE',52),('SUR',53),('NORTE',53),('ESTE',53),('OESTE',53),('SUR',54),('NORTE',54),('ESTE',54),('OESTE',54),('SUR',55),('NORTE',55),('ESTE',55),('OESTE',55),('SUR',56),('NORTE',56),('ESTE',56),('OESTE',56),('SUR',57),('NORTE',57),('ESTE',57),('OESTE',57),('SUR',58),('NORTE',58),('ESTE',58),('OESTE',58),('SUR',59),('NORTE',59),('ESTE',59),('OESTE',59),('SUR',60),('NORTE',60),('ESTE',60),('OESTE',60),('SUR',61),('NORTE',61),('ESTE',61),('OESTE',61),('SUR',62),('NORTE',62),('ESTE',62),('OESTE',62),('SUR',63),('NORTE',63),('ESTE',63),('OESTE',63),('SUR',64),('NORTE',64),('ESTE',64),('OESTE',64),('SUR',65),('NORTE',65),('ESTE',65),('OESTE',65),('SUR',66),('NORTE',66),('ESTE',66),('OESTE',66),('SUR',67),('NORTE',67),('ESTE',67),('OESTE',67),('SUR',68),('NORTE',68),('ESTE',68),('OESTE',68),('SUR',69),('NORTE',69),('ESTE',69),('OESTE',69),('SUR',70),('NORTE',70),('ESTE',70),('OESTE',70),('SUR',71),('NORTE',71),('ESTE',71),('OESTE',71),('SUR',72),('NORTE',72),('ESTE',72),('OESTE',72),('SUR',73),('NORTE',73),('ESTE',73),('OESTE',73),('SUR',74),('NORTE',74),('ESTE',74),('OESTE',74),('SUR',75),('NORTE',75),('ESTE',75),('OESTE',75),('SUR',76),('NORTE',76),('ESTE',76),('OESTE',76),('SUR',77),('NORTE',77),('ESTE',77),('OESTE',77),('SUR',78),('NORTE',78),('ESTE',78),('OESTE',78),('SUR',79),('NORTE',79),('ESTE',79),('OESTE',79),('SUR',80),('NORTE',80),('ESTE',80),('OESTE',80),('SUR',81),('NORTE',81),('ESTE',81),('OESTE',81),('SUR',82),('NORTE',82),('ESTE',82),('OESTE',82),('SUR',83),('NORTE',83),('ESTE',83),('OESTE',83),('SUR',84),('NORTE',84),('ESTE',84),('OESTE',84),('SUR',85),('NORTE',85),('ESTE',85),('OESTE',85),('SUR',86),('NORTE',86),('ESTE',86),('OESTE',86),('SUR',87),('NORTE',87),('ESTE',87),('OESTE',87),('SUR',88),('NORTE',88),('ESTE',88),('OESTE',88),('SUR',89),('NORTE',89),('ESTE',89),('OESTE',89),('SUR',90),('NORTE',90),('ESTE',90),('OESTE',90),('SUR',91),('NORTE',91),('ESTE',91),('OESTE',91),('SUR',92),('NORTE',92),('ESTE',92),('OESTE',92),('SUR',93),('NORTE',93),('ESTE',93),('OESTE',93),('SUR',94),('NORTE',94),('ESTE',94),('OESTE',94),('SUR',95),('NORTE',95),('ESTE',95),('OESTE',95),('SUR',96),('NORTE',96),('ESTE',96),('OESTE',96),('SUR',97),('NORTE',97),('ESTE',97),('OESTE',97),('SUR',98),('NORTE',98),('ESTE',98),('OESTE',98),('SUR',99),('NORTE',99),('ESTE',99),('OESTE',99),('SUR',100),('NORTE',100),('ESTE',100),('OESTE',100),('SUR',101),('NORTE',101),('ESTE',101),('OESTE',101),('SUR',102),('NORTE',102),('ESTE',102),('OESTE',102),('SUR',103),('NORTE',103),('ESTE',103),('OESTE',103),('SUR',104),('NORTE',104),('ESTE',104),('OESTE',104),('SUR',105),('NORTE',105),('ESTE',105),('OESTE',105),('SUR',106),('NORTE',106),('ESTE',106),('OESTE',106),('SUR',107),('NORTE',107),('ESTE',107),('OESTE',107),('SUR',108),('NORTE',108),('ESTE',108),('OESTE',108),('SUR',109),('NORTE',109),('ESTE',109),('OESTE',109),('SUR',110),('NORTE',110),('ESTE',110),('OESTE',110),('SUR',111),('NORTE',111),('ESTE',111),('OESTE',111),('SUR',112),('NORTE',112),('ESTE',112),('OESTE',112),('SUR',113),('NORTE',113),('ESTE',113),('OESTE',113),('SUR',114),('NORTE',114),('ESTE',114),('OESTE',114),('SUR',115),('NORTE',115),('ESTE',115),('OESTE',115),('SUR',116),('NORTE',116),('ESTE',116),('OESTE',116),('SUR',117),('NORTE',117),('ESTE',117),('OESTE',117),('SUR',118),('NORTE',118),('ESTE',118),('OESTE',118),('SUR',119),('NORTE',119),('ESTE',119),('OESTE',119),('SUR',120),('NORTE',120),('ESTE',120),('OESTE',120),('SUR',121),('NORTE',121),('ESTE',121),('OESTE',121),('SUR',122),('NORTE',122),('ESTE',122),('OESTE',122),('SUR',123),('NORTE',123),('ESTE',123),('OESTE',123),('SUR',124),('NORTE',124),('ESTE',124),('OESTE',124),('SUR',125),('NORTE',125),('ESTE',125),('OESTE',125),('SUR',126),('NORTE',126),('ESTE',126),('OESTE',126),('SUR',127),('NORTE',127),('ESTE',127),('OESTE',127),('SUR',128),('NORTE',128),('ESTE',128),('OESTE',128),('SUR',129),('NORTE',129),('ESTE',129),('OESTE',129),('SUR',130),('NORTE',130),('ESTE',130),('OESTE',130),('SUR',131),('NORTE',131),('ESTE',131),('OESTE',131),('SUR',132),('NORTE',132),('ESTE',132),('OESTE',132),('SUR',133),('NORTE',133),('ESTE',133),('OESTE',133),('SUR',134),('NORTE',134),('ESTE',134),('OESTE',134),('SUR',135),('NORTE',135),('ESTE',135),('OESTE',135),('SUR',136),('NORTE',136),('ESTE',136),('OESTE',136);

insert into personas (nombre,apellido,documento,contacto,dire\_persona,id\_barrio) VALUES

('Curran','Riddle',39685621,3515008249,'9872 Elit. Avenue',241),

('Vance','Witt',39688060,3515009302,'Ap #769-6603 Etiam Road',429),

('Lani','Wilkinson',39690499,3515010355,'Ap #777-5111 Justo Rd.',14),

('Macey','Macdonald',39692938,3515011408,'Ap #455-117 Aliquet Street',14),

('Steven','Stokes',39695377,3515012461,'P.O. Box 692, 936 Sociis Rd.',345),

('Flynn','Holland',39697816,3515013514,'1172 Quam. Av.',430),

('Leo','Beard',39700255,3515014567,'199-3752 Sit Avenue',358),

('Jamalia','Holder',39702694,3515015620,'Ap #965-6512 Euismod Road',85),

('Delilah','Lambert',39705133,3515016673,'6559 Aliquam Ave',309),

('Amena','O''connor',39707572,3515017726,'266-5632 Semper Road',178),

('Aurelia','David',39710011,3515018779,'4598 Sed Avenue',265),

('Maggy','Wooten',39712450,3515019832,'4017 Consequat, Ave',125),

('Keelie','Romero',39714889,3515020885,'234-791 Pharetra. Rd.',494),

('Ciaran','Barton',39717328,3515021938,'363-3556 Scelerisque Street',166),

('Herrod','Conley',39719767,3515022991,'2051 Ridiculus Av.',41),

('Juliet','Hoffman',39722206,3515024044,'2755 Dapibus Rd.',114),

('Yetta','Webb',39724645,3515025097,'424-740 Duis Av.',492),

('Marcia','Best',39727084,3515026150,'Ap #463-844 Viverra. St.',226),

('Madison','Gallegos',39729523,3515027203,'401-2342 Ornare, Rd.',203),

('Keiko','Dixon',39731962,3515028256,'Ap #267-6351 Suspendisse Street',466),

('Edward','Fry',39734401,3515029309,'7835 Nibh. Street',123),

('Warren','Hess',39736840,3515030362,'Ap #217-9369 Praesent Avenue',417),

('Jermaine','Alford',39739279,3515031415,'Ap #507-5862 Per Av.',479),

('Rhea','Buckley',39741718,3515032468,'112-2633 Quisque Avenue',289),

('Madaline','Hunt',39744157,3515033521,'728-477 Amet Street',147),

('Connor','Collins',39746596,3515034574,'631-7864 Eget St.',290),

('Urielle','Rodriguez',39749035,3515035627,'Ap #307-5143 Luctus Rd.',219),

('MacKenzie','Hardin',39751474,3515036680,'Ap #109-2066 Sit Ave',455),

('Jada','Clemons',39753913,3515037733,'156-3407 Leo. Ave',56),

('Colleen','Page',39756352,3515038786,'Ap #837-8520 Etiam Street',93),

('Charde','Petersen',39758791,3515039839,'749 Felis, Street',291),

('Keiko','Graham',39761230,3515040892,'7176 Cubilia Av.',361),

('Mia','Erickson',39763669,3515041945,'317-1332 Non, Road',479),

('Kenneth','Buckley',39766108,3515042998,'156-2689 Nec Rd.',522),

('Keegan','Hunt',39768547,3515044051,'808-3213 Vestibulum, Rd.',398),

('Baker','Campbell',39770986,3515045104,'Ap #824-823 Parturient St.',244),

('Adele','Mccarthy',39773425,3515046157,'5749 Elit, St.',141),

('MacKenzie','Carlson',39775864,3515047210,'259-1632 Ut Avenue',147),

('Colin','Flores',39778303,3515048263,'9599 Euismod St.',277),

('William','Thomas',39780742,3515049316,'Ap #747-8199 Penatibus St.',283),

('Samuel','Michael',39783181,3515050369,'686-1358 Vitae, Road',386),

('Molly','Mclean',39785620,3515051422,'Ap #794-7305 Eleifend. St.',126),

('Victor','Cherry',39788059,3515052475,'3678 Felis. Rd.',268),

('Evelyn','Slater',39790498,3515053528,'P.O. Box 923, 4488 Justo. Rd.',223),

('Colleen','Lambert',39792937,3515054581,'856-7178 Orci Rd.',354),

('Merritt','Christian',39795376,3515055634,'Ap #251-5336 Enim, Rd.',432),

('Kaye','Kelley',39797815,3515056687,'7490 At Road',543),

('Arthur','Rodriguez',39800254,3515057740,'Ap #834-9927 Blandit Rd.',197),

('Dominic','Roach',39802693,3515058793,'Ap #976-592 Vel, Street',153),

('Zachery','Newton',39805132,3515059846,'972-8810 Nec St.',401),

('Keely','Franco',39807571,3515060899,'314-1298 Scelerisque Road',132),

('Sylvia','Skinner',39810010,3515061952,'P.O. Box 316, 339 Neque Rd.',90),

('Emily','Tran',39812449,3515063005,'Ap #865-2859 Dui. Av.',433),

('Rhoda','Gross',39814888,3515064058,'Ap #359-3103 Ligula. Ave',261),

('Jerry','Klein',39817327,3515065111,'4618 Fusce Avenue',198),

('Devin','Duke',39819766,3515066164,'Ap #664-6432 Purus, Av.',170),

('Addison','Cantu',39822205,3515067217,'2791 Augue Av.',469),

('Germaine','House',39824644,3515068270,'Ap #809-9397 Elementum Street',179),

('Uriah','Thompson',39827083,3515069323,'Ap #838-7472 Velit Road',490),

('Tucker','Hahn',39829522,3515070376,'Ap #467-1606 Lorem, St.',109);

insert into tipo\_clientes (tipo\_cliente,razon\_social,dire\_empresa,cuit,contacto,descuento,id\_barrio) VALUES

('AUTOPARTES','Feugiat Lorem Company','2223 Lectus. Road',30450819405,'duis.mi@protonmail.edu',10.00,329),

('AUTOPARTES','Pellentesque Ltd','116-2554 Lacus. Ave',30412798436,'ac.risus.morbi@google.ca',10.00,46),

('AUTOPARTES','Donec Industries','P.O. Box 287, 1774 Erat Road',30498897083,'ante.lectus.convallis@icloud.ca',10.00,44),

('AUTOPARTES','Iaculis Corporation','6024 Vestibulum Ave',30498174844,'scelerisque@hotmail.edu',10.00,219),

('AUTOPARTES','Nunc Ullamcorper Limited','923-5517 Eleifend St.',30459954721,'curabitur.massa@protonmail.net',10.00,433),

('AUTOPARTES','Mauris Morbi Corporation','Ap #879-9113 Purus Street',30592234688,'lacus.quisque@icloud.com',10.00,10),

('AUTOPARTES','Vestibulum Ante Inc.','5893 Risus Av.',30605604298,'sociosqu.ad.litora@google.ca',10.00,408),

('AUTOPARTES','Sapien Cursus In Associates','2193 Felis. Rd.',30448134536,'hendrerit.id@google.edu',10.00,393),

('AUTOPARTES','Tellus Aenean Associates','8210 Malesuada Street',30578121171,'arcu.ac@aol.net',10.00,54),

('AUTOPARTES','Risus Donec Institute','P.O. Box 423, 670 Nibh. St.',30600281073,'mattis.semper.dui@google.org',10.00,137),

('CONSUMIDOR FINAL','Blandit Corporation','P.O. Box 690, 1650 Turpis. Rd.',30411693583,'massa.rutrum@yahoo.org',00.00,172),

('CONSUMIDOR FINAL','Semper Rutrum Fusce Foundation','P.O. Box 233, 9975 Quis Road',30564992284,'eros.turpis.non@protonmail.edu',00.00,203),

('CONSUMIDOR FINAL','Lorem Corporation','Ap #976-491 Interdum. Rd.',30590640940,'mauris.morbi@outlook.com',00.00,19),

('CONSUMIDOR FINAL','Faucibus Orci Luctus Institute','Ap #927-2305 Ante. Rd.',30575481995,'mattis.semper.dui@protonmail.org',00.00,341),

('CONSUMIDOR FINAL','Aliquam Adipiscing Corp.','P.O. Box 103, 7272 Vel, Street',30456045112,'cum@google.com',00.00,423),

('CONSUMIDOR FINAL','Cras Limited','P.O. Box 363, 9318 Sagittis Road',30505568369,'diam.eu.dolor@google.couk',00.00,280),

('CONSUMIDOR FINAL','Nisi Nibh Lacinia Limited','219-8219 Mauris, Ave',30437956709,'nisi.magna@hotmail.net',00.00,174),

('CONSUMIDOR FINAL','Nec Mollis Vitae Institute','701-4727 Elementum, St.',30519633289,'libero.lacus.varius@aol.org',00.00,397),

('CONSUMIDOR FINAL','Lectus Convallis Inc.','927-4245 Aliquam Road',30401402859,'nam.ac.nulla@protonmail.com',00.00,211),

('CONSUMIDOR FINAL','Mollis Phasellus Corporation','P.O. Box 199, 7350 In Rd.',30521180689,'arcu@icloud.couk',00.00,211),

('CONCESIONARIO','Netus Associates','Ap #847-1099 Sagittis. Av.',30501204446,'interdum@hotmail.net',00.00,282),

('CONCESIONARIO','Tempus Risus Donec Corporation','P.O. Box 102, 7757 Nec, Ave',30552869505,'eget@icloud.net',00.00,113),

('CONCESIONARIO','Nisl Nulla Foundation','1450 Vel, Avenue',30472537688,'dictum.ultricies@hotmail.couk',00.00,215),

('CONCESIONARIO','Nibh Enim LLP','Ap #464-1900 Congue, Ave',30604854315,'velit.cras@yahoo.org',00.00,136),

('CONCESIONARIO','Erat Sed Nunc LLP','Ap #620-3505 Etiam Road',30557349949,'aliquam.nisl@outlook.org',00.00,355),

('CONCESIONARIO','Ac Orci Ut Associates','Ap #711-9212 Ipsum St.',30440280064,'magnis.dis@outlook.org',00.00,534),

('CONCESIONARIO','Vel Pede Blandit Corporation','938-9082 Metus Rd.',30449308842,'nec.mauris@yahoo.net',00.00,162),

('CONCESIONARIO','Vel Faucibus Id Ltd','P.O. Box 566, 3430 Sem St.',30457221024,'diam.sed@outlook.couk',00.00,422),

('CONCESIONARIO','Ornare In Faucibus Corporation','P.O. Box 630, 1859 Curabitur Rd.',30440798863,'quis.lectus@yahoo.com',00.00,64),

('CONCESIONARIO','Elementum LLP','6459 Dolor Rd.',30462995897,'in.faucibus@outlook.ca',00.00,486),

('EMPRESA','Quam Pellentesque LLC','437-8609 Amet Ave',30542967636,'volutpat.nulla@outlook.org',00.00,179),

('EMPRESA','Pede Cum Sociis Incorporated','P.O. Box 839, 7520 Ipsum St.',30570115999,'nec.ligula.consectetuer@aol.org',00.00,131),

('EMPRESA','Lacinia Vitae Sodales Incorporated','6908 Est. St.',30572027877,'lorem.auctor.quis@aol.com',00.00,15),

('EMPRESA','Aliquam Erat Company','7441 Eu St.',30470414382,'nisl.maecenas.malesuada@hotmail.ca',00.00,429),

('EMPRESA','Venenatis LLP','Ap #291-2511 Scelerisque Rd.',30428436285,'amet.ante@hotmail.edu',00.00,85),

('EMPRESA','Donec Luctus Company','296-136 Cursus. Avenue',30577073270,'vitae.erat@hotmail.edu',00.00,476),

('EMPRESA','Cursus Limited','4070 Hendrerit Road',30515843479,'tincidunt@yahoo.couk',00.00,509),

('EMPRESA','Vitae Purus Gravida Institute','175-6851 Scelerisque, Rd.',30538701033,'nascetur.ridiculus.mus@outlook.couk',00.00,95),

('EMPRESA','Magna Cras PC','P.O. Box 374, 1163 Tellus St.',30466853880,'nisl.quisque.fringilla@google.ca',00.00,223),

('EMPRESA','Ante Foundation','337-4537 Scelerisque Av.',30461512609,'enim.gravida@icloud.net',00.00,91);

insert into clientes (id\_persona,id\_tipo\_cliente) VALUES (1,40),(2,39),(3,38),(4,37),(5,36),(6,35),(7,34),(8,33),(9,32),(10,31),(11,30),(12,29),(13,28),(14,27),(15,26),(16,25),(17,24),(18,23),(19,22),(20,21),(21,20),(22,19),(23,18),(24,17),(25,16),(26,15),(27,14),(28,13),(29,12),(30,11),(31,10),(32,9),(33,8),(34,7),(35,6),(36,5),(37,4),(38,3),(39,2),(40,1);

insert into tipo\_empleados (tipo\_empleado) VALUES ('empleado');

insert into empleados (id\_persona,id\_tipo\_empleado,cuit) VALUES (41,1,20434861463),(42,1,20398221538),(43,1,20402965297),(44,1,20432722908),(45,1,20396748862),(46,1,20409904167),(47,1,20411552925),(48,1,20391771875),(49,1,20422428108),(50,1,20414885053);

insert into tipo\_productos (tipo\_producto) VALUES ('autoparte'),('vehiculo');

insert into marcas (marca) VALUES ('Audi'),('BMW'),('Cadillac'),('Chevrolet'),('Citroen'),('Dacia'),('Ferrari'),('Fiat'),('Ford'),('Honda');

insert into productos (descripcion,anio,stock,precio\_vta,id\_marca,id\_tipo\_producto) VALUES

('baul',2010,1,1200,1,1),

('baul',2020,1,500,2,1),

('baul',2020,1,1500,3,1),

('baul',2014,1,600,4,1),

('baul',2018,1,1900,5,1),

('baul',2018,0,200,6,1),

('baul',2019,0,2000,7,1),

('baul',2011,0,700,8,1),

('baul',2011,0,400,9,1),

('baul',2017,0,1600,10,1),

('capot',2016,1,1000,1,1),

('capot',2012,1,1800,2,1),

('capot',2013,1,100,3,1),

('capot',2019,1,1100,4,1),

('capot',2018,1,1400,5,1),

('capot',2019,0,300,6,1),

('capot',2012,0,1300,7,1),

('capot',2019,0,800,8,1),

('capot',2015,0,900,9,1),

('capot',2012,0,1700,10,1),

('carroceria',2018,1,1200,1,1),

('carroceria',2011,1,500,2,1),

('carroceria',2015,1,1500,3,1),

('carroceria',2014,1,600,4,1),

('carroceria',2018,1,1900,5,1),

('carroceria',2017,0,200,6,1),

('carroceria',2013,0,2000,7,1),

('carroceria',2020,0,700,8,1),

('carroceria',2017,0,400,9,1),

('carroceria',2018,0,1600,10,1),

('freno',2016,1,1000,1,1),

('freno',2016,1,1800,2,1),

('freno',2013,1,100,3,1),

('freno',2019,1,1100,4,1),

('freno',2020,1,1400,5,1),

('freno',2020,0,300,6,1),

('freno',2012,0,1300,7,1),

('freno',2019,0,800,8,1),

('freno',2020,0,900,9,1),

('freno',2016,0,1700,10,1),

('luneta',2016,1,1200,1,1),

('luneta',2018,1,500,2,1),

('luneta',2019,1,1500,3,1),

('luneta',2013,1,600,4,1),

('luneta',2016,1,1900,5,1),

('luneta',2020,0,200,6,1),

('luneta',2010,0,2000,7,1),

('luneta',2019,0,700,8,1),

('luneta',2013,0,400,9,1),

('luneta',2014,0,1600,10,1),

('motor',2020,1,1000,1,1),

('motor',2012,1,1800,2,1),

('motor',2011,1,100,3,1),

('motor',2018,1,1100,4,1),

('motor',2021,1,1400,5,1),

('motor',2012,0,300,6,1),

('motor',2011,0,1300,7,1),

('motor',2020,0,800,8,1),

('motor',2015,0,900,9,1),

('motor',2018,0,1700,10,1),

('puerta',2010,1,1200,1,1),

('puerta',2012,1,500,2,1),

('puerta',2020,1,1500,3,1),

('puerta',2016,1,600,4,1),

('puerta',2020,1,1900,5,1),

('puerta',2015,0,200,6,1),

('puerta',2010,0,2000,7,1),

('puerta',2018,0,700,8,1),

('puerta',2017,0,400,9,1),

('puerta',2010,0,1600,10,1),

('parabrisa',2013,1,1000,1,1),

('parabrisa',2017,1,1800,2,1),

('parabrisa',2016,1,100,3,1),

('parabrisa',2013,1,1100,4,1),

('parabrisa',2013,1,1400,5,1),

('parabrisa',2018,0,300,6,1),

('parabrisa',2011,0,1300,7,1),

('parabrisa',2017,0,800,8,1),

('parabrisa',2010,0,900,9,1),

('parabrisa',2013,0,1700,10,1),

('rueda',2015,1,1200,1,1),

('rueda',2014,1,500,2,1),

('rueda',2011,1,1500,3,1),

('rueda',2014,1,600,4,1),

('rueda',2015,1,1900,5,1),

('rueda',2011,0,200,6,1),

('rueda',2013,0,2000,7,1),

('rueda',2012,0,700,8,1),

('rueda',2015,0,400,9,1),

('rueda',2013,0,1600,10,1),

('ventanilla',2018,1,1000,1,1),

('ventanilla',2019,1,1800,2,1),

('ventanilla',2012,1,100,3,1),

('ventanilla',2014,1,1100,4,1),

('ventanilla',2015,1,1400,5,1),

('ventanilla',2016,0,300,6,1),

('ventanilla',2021,0,1300,7,1),

('ventanilla',2014,0,800,8,1),

('ventanilla',2011,0,900,9,1),

('ventanilla',2014,0,1700,10,1);

insert into productos (descripcion,anio,stock,precio\_vta,id\_marca,id\_tipo\_producto) VALUES ('3 puertas',2022,0,20000,1,2),('5 puertas',2022,0,30000,2,2),('3 puertas',2022,0,10000,3,2),('5 puertas',2022,0,40000,4,2),('3 puertas',2022,0,15000,5,2),('5 puertas',2022,0,25000,6,2),('3 puertas',2022,0,35000,7,2),('5 puertas',2022,0,20000,8,2),('3 puertas',2022,0,30000,9,2),('5 puertas',2022,0,10000,10,2)

insert into metodo\_pagos(metodo\_pago) VALUES ('efectivo'),('tarjeta'),('autoplan'),('transferencia');

insert into tipo\_envios (tipo\_envio) VALUES ('domicilio'),('retiro');

SET DATEFORMAT DMY;

insert into facturas (fecha\_factura,id\_metodo\_pago,id\_tipo\_envio) VALUES

('02-09-2022',2,1),

('01-09-2022',2,1),

('12-09-2022',1,1),

('17-09-2022',1,1),

('30-09-2022',3,2),--autoplan

('13-08-2022',2,1),

('08-09-2022',3,2),--autoplan

('05-08-2022',2,1),

('07-11-2022',2,1),

('09-09-2022',4,1),

('07-09-2022',4,1),

('23-08-2022',4,2),

('29-09-2022',1,2);

insert into pedidos (id\_factura, id\_empleado,id\_cliente,fecha\_pedido,fecha\_entrega) VALUES

(null,1,1,'30-08-2022','22-12-2022'),

(null,2,2,'16-09-2022','30-10-2022'),

(1,3,3,'07-08-2022','02-09-2022'),--

(2,4,4,'22-08-2022','01-09-2022'),--

(null,5,5,'30-07-2022','20-10-2022'),

(null,6,6,'24-07-2022','06-10-2022'),

(null,7,7,'12-09-2022','30-11-2022'),

(null,8,8,'27-08-2022','14-09-2022'),

(null,9,9,'30-06-2022','17-11-2022'),

(null,10,10,'14-08-2022','21-09-2022'),

(null,1,11,'10-07-2022','05-10-2022'),

(null,2,12,'23-08-2022','07-10-2022'),

(null,3,13,'08-09-2022','09-11-2022'),

(3,4,14,'05-08-2022','12-09-2022'),--

(4,5,15,'08-06-2022','17-09-2022'),--

(null,6,16,'23-06-2022','14-09-2022'),-----no se entrego

(null,7,17,'26-08-2022','17-09-2022'),------ no se entrego

(null,8,18,'03-06-2022','03-10-2022'),

(null,9,19,'12-10-2022','15-11-2022'),

(null,10,20,'08-09-2022','28-12-2022'),

(5,1,21,'04-06-2022','30-09-2022'),--

(6,2,22,'13-08-2022','13-08-2022'),--mismo dia

(7,3,23,'18-08-2022','08-09-2022'),--

(8,4,24,'05-08-2022','05-08-2022'),--mismo dia

(9,5,25,'07-11-2022','07-11-2022'),--mismo dia

(null,6,26,'06-08-2022','24-12-2022'),

(null,7,27,'25-11-2022','01-11-2022'),

(null,8,28,'12-09-2022','27-10-2022'),

(null,9,29,'05-10-2022','04-12-2022'),

(null,10,30,'25-10-2022','07-10-2022'),

(null,1,31,'13-09-2022','14-11-2022'),

(null,2,32,'18-10-2022','25-10-2022'),

(10,3,33,'09-08-2022','09-09-2022'),--

(null,4,34,'22-09-2022','01-11-2022'),

(11,5,35,'06-06-2022','07-09-2022'),--

(null,6,36,'10-10-2022','15-09-2022'),----no se entrego

(12,7,37,'23-08-2022','23-08-2022'),--mismo dia

(13,8,38,'29-09-2022','29-09-2022'),--mismo dia

(null,9,39,'13-08-2022','03-12-2022'),

(null,10,40,'11-09-2022','06-11-2022');

insert into detalles (id\_factura,id\_pedido,id\_producto,precio\_unitario,cantidad) VALUES

(1,4,101,20000,1),

(1,4,102,30000,1),

(1,4,103,10000,1),

(1,4,104,40000,1),

(1,4,105,15000,1),

(2,5,106,25000,1),

(2,5,107,35000,1),

(2,5,108,20000,1),

(2,5,109,30000,1),

(2,5,110,10000,1),

(3,14,109,30000,100),

(4,15,107,35000,50),

(5,21,104,40000,1),--autoplan

(6,22,76,300,2),

(7,23,110,10000,1),--autoplan

(8,24,81,1200,1),

(9,25,55,1400,1),

(10,33,35,1400,15),

(11,35,38,800,7),

(12,37,42,500,3),

(13,38,45,1900,42);