Proyecto SQL

Feria Placarcito

# Introducción

## Descripción de la Situación de Negocio

En la feria placarcito, nos dedicamos a facilitar la venta de productos nuevos y usados en una variedad de categorías, desde productos electrónicos hasta ropa libros, juguetes y zapatos. Además, ofrecemos un excelente servicio post venta brindándole al usuario la posibilidad de devolver el producto dentro de los 30 días de entregado, sea por el motivo que decida. Ofrecemos calificaciones y reseñas de usuarios, para fomentar la confianza y la transparencia en las transacciones.

## Objetivo de la Base de Datos

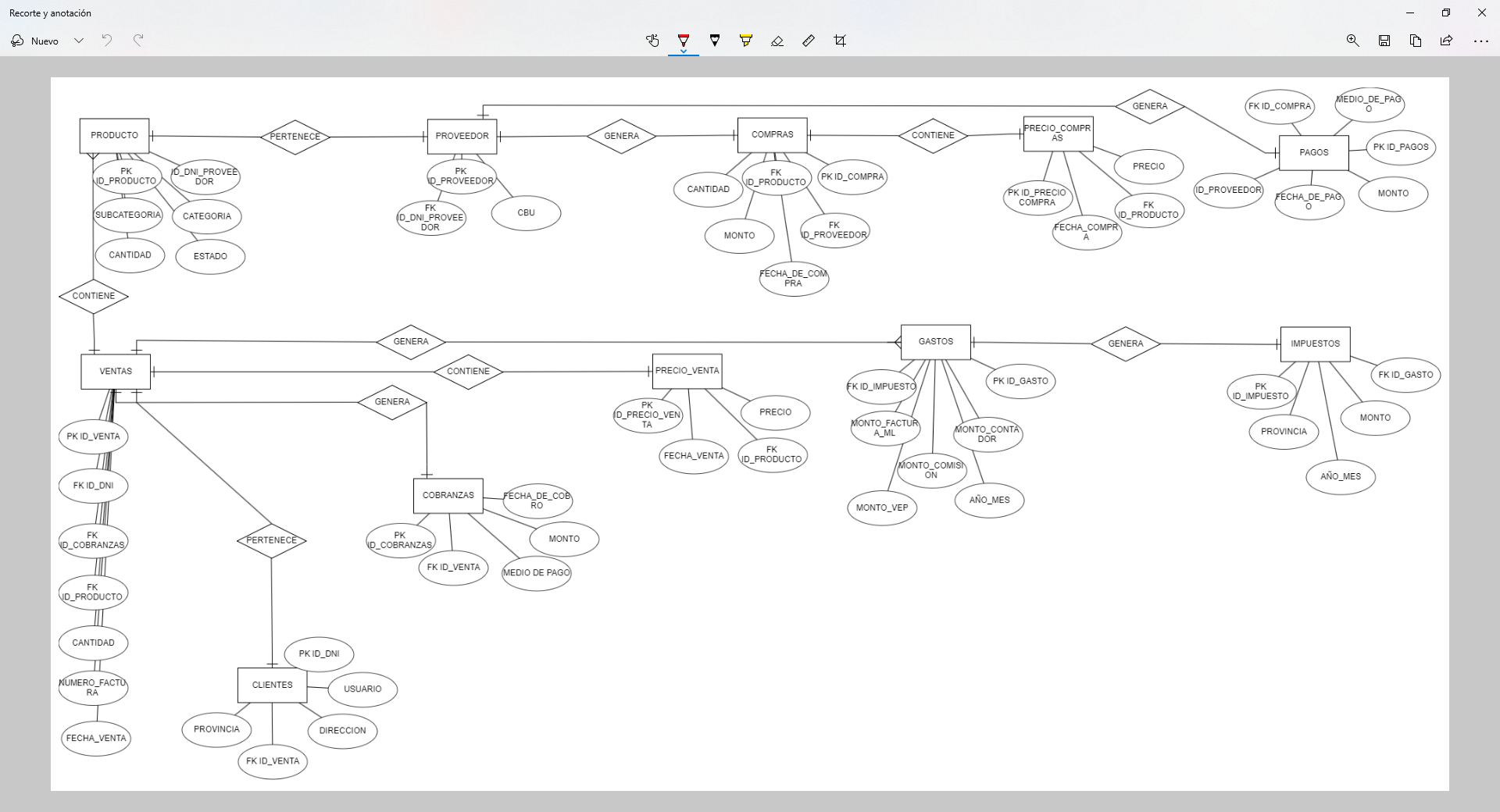
* Almacenar información detallada de los clientes, incluyendo datos personales e información de contacto.
* Mantener un registro completo de los productos disponibles para la venta, incluyendo descripciones, precios y detalles relevantes.
* Mantener un registro completo de los proveedores de los productos junto con el detalle de la compra de los mismos, la fecha de compra y el precio de compra.
* Registrar las transacciones realizadas, incluyendo la fecha, el detalle de los productos vendidos, el precio de venta y el lugar de venta.
* Almacenar calificaciones y reseñas de usuarios para garantizar la confiabilidad de los vendedores y fomentar la transparencia en la plataforma.

## Problemas a resolver

1. Gestión Eficiente de la Información: Con la incorporación constante de productos, es esencial mantener y acceder a la información de manera eficiente. La base de datos permitirá una gestión eficaz del negocio. Teniendo organizado mediante códigos de identificación a quien se le compró determinado producto (proveedor) y luego a quien se le vendió mediante la plataforma.
2. Facilitación de Transacciones: Facilitará la realización de transacciones, asegurando que los usuarios puedan buscar productos, realizar compras y reclamos de manera rápida y efectiva.
3. Confiabilidad y Transparencia: Fomentará la confiabilidad mediante la gestión de calificaciones y reseñas, lo que ayudará a tomar decisiones informadas al comprar productos y en consecuencia hacer crecer el negocio ya que cada reseña positiva suma para pasar de Mercado Líder a Mercado Platinum.
4. Escalabilidad: La base de datos debe ser escalable para adaptarse al crecimiento del negocio a lo largo del tiempo sin comprometer el rendimiento. Actualmente cuenta con 1500 productos la idea del negocio es superar los 10.000

# Entidades y relaciones

## Diagrama Entidad-Relación



## Descripción de tablas

### Tabla: Producto

Esta tabla almacena información sobre los productos disponibles de venta en Placarcito.



### Tabla: Proveedor

Esta tabla almacena información de los proveedores de placarcito.

### 

### Tabla: Clientes

Esta tabla registra los datos de los clientes que compran en placarcito.

### 

### Tabla: Cobranzas

Esta tabla contiene información de los cobros recibidos por cada venta realizada.

### Tabla: Pagos

Esta tabla contiene la información de los pagos realizados a cada proveedor.



### Tabla: Compras

Esta tabla contiene información de los productos comprados al proveedor para luego ser vendidos en placarcito.

### 

### Tabla: Precio de Compras

Esta tabla contiene información de los precios de cada producto comprado para luego ser vendido en placarcito.

### 

### Tabla: Precio de Ventas

Esta tabla contiene los precios de los productos vendidos desde la plataforma.

### 

### Tabla: Gastos

Esta tabla registra los gastos mensuales que genera la feria.



### Tabla: Ventas

Esta tabla registra la información de cada venta realizada en la feria.



### Tabla: Impuestos

Esta tabla registra la información cada impuesto cobrado en cada provincia donde se vende.



# Scripts

## Creación de tablas

-- Crear la base de datos de placarcito

DROP SCHEMA IF EXISTS FERIA\_PLACARCITO ;

CREATE SCHEMA IF NOT EXISTS FERIA\_PLACARCITO ;

USE FERIA\_PLACARCITO ;

CREATE TABLE PRODUCTO (

ID\_PRODUCTO INT AUTO\_INCREMENT,

CANTIDAD INT,

ID\_DNI\_PROVEEDOR INT,

CATEGORIA VARCHAR (50),

SUBCATEGORIA VARCHAR (50),

ESTADO ENUM ('NUEVO', 'USADO') NOT NULL,

PRIMARY KEY (ID\_PRODUCTO)

);

CREATE TABLE PROVEEDOR (

ID\_PROVEEDOR INT NOT NULL AUTO\_INCREMENT,

CBU VARCHAR (50),

ID\_PRODUCTO INT,

ID\_DNI\_PROVEEDOR INT,

PRIMARY KEY (ID\_PROVEEDOR)

);

CREATE TABLE CLIENTES (

ID\_DNI INT,

DIRECCION VARCHAR (50),

USUARIO VARCHAR (50),

PROVINCIA VARCHAR (50),

PRIMARY KEY (ID\_DNI)

);

CREATE TABLE COBRANZAS (

ID\_COBRANZAS INT NOT NULL AUTO\_INCREMENT,

MONTO FLOAT,

FECHA\_DE\_COBRO TIMESTAMP,

MEDIO\_PAGO VARCHAR (50),

ID\_VENTA INT NOT NULL,

PRIMARY KEY (ID\_COBRANZAS)

);

CREATE TABLE PAGOS (

ID\_PAGOS INT NOT NULL AUTO\_INCREMENT,

ID\_COMPRA INT NOT NULL,

ID\_PROVEEDOR INT NOT NULL,

MONTO FLOAT,

FECHA\_DE\_PAGO DATE,

MEDIO\_PAGO VARCHAR (50),

PRIMARY KEY (ID\_PAGOS)

);

CREATE TABLE COMPRAS (

ID\_COMPRA INT NOT NULL AUTO\_INCREMENT,

CANTIDAD INT,

ID\_PRODUCTO INT NOT NULL,

ID\_PROVEEDOR INT NOT NULL,

MONTO FLOAT,

FECHA\_COMPRA DATE,

PRIMARY KEY (ID\_COMPRA)

);

CREATE TABLE PRECIO\_COMPRAS (

ID\_PRECIO\_COMPRA INT NOT NULL AUTO\_INCREMENT,

PRECIO FLOAT,

FECHA\_COMPRA DATE,

ID\_PRODUCTO INT NOT NULL,

PRIMARY KEY (ID\_PRECIO\_COMPRA)

);

CREATE TABLE PRECIO\_VENTA (

ID\_PRECIO\_VENTA INT NOT NULL AUTO\_INCREMENT,

PRECIO FLOAT,

ID\_PRODUCTO INT NOT NULL,

FECHA\_VENTA DATE,

PRIMARY KEY (ID\_PRECIO\_VENTA)

);

CREATE TABLE GASTOS (

ID\_GASTO INT NOT NULL AUTO\_INCREMENT,

MONTO\_VEP FLOAT,

MONTO\_CONTADOR FLOAT,

MONTO\_FACTURA\_ML FLOAT,

AÑO\_MES DATE,

MONTO\_COMISION\_ML FLOAT,

PRIMARY KEY (ID\_GASTO)

);

CREATE TABLE VENTAS (

ID\_VENTA INT NOT NULL AUTO\_INCREMENT,

ID\_PRODUCTO INT NOT NULL,

CANTIDAD INT,

FECHA\_DE\_VENTA TIMESTAMP,

NUMERO\_FACTURA INT,

ID\_DNI INT,

PRIMARY KEY (ID\_VENTA)

);

CREATE TABLE IMPUESTOS (

ID\_IMPUESTO INT NOT NULL AUTO\_INCREMENT,

ID\_GASTO INT NOT NULL,

PROVINCIA VARCHAR (50),

MONTO FLOAT,

AÑO\_MES DATE,

PRIMARY KEY (ID\_IMPUESTO)

);

ALTER TABLE VENTAS

ADD FOREIGN KEY (ID\_DNI) REFERENCES CLIENTES (ID\_DNI),

ADD FOREIGN KEY (ID\_PRODUCTO) REFERENCES PRODUCTO (ID\_PRODUCTO);

ALTER TABLE COMPRAS

ADD FOREIGN KEY (ID\_PRODUCTO) REFERENCES PRODUCTO (ID\_PRODUCTO),

ADD FOREIGN KEY (ID\_PROVEEDOR) REFERENCES PROVEEDOR (ID\_PROVEEDOR);

ALTER TABLE PRECIO\_COMPRAS

ADD FOREIGN KEY (ID\_PRODUCTO) REFERENCES PRODUCTO (ID\_PRODUCTO);

ALTER TABLE PRECIO\_VENTA

ADD FOREIGN KEY (ID\_PRODUCTO) REFERENCES PRODUCTO (ID\_PRODUCTO);

ALTER TABLE COBRANZAS

ADD FOREIGN KEY (ID\_VENTA) REFERENCES VENTAS (ID\_VENTA);

ALTER TABLE PAGOS

ADD FOREIGN KEY (ID\_COMPRA) REFERENCES COMPRAS (ID\_COMPRA);

ALTER TABLE IMPUESTOS

ADD FOREIGN KEY (ID\_GASTO) REFERENCES GASTOS (ID\_GASTO);

ALTER TABLE PROVEEDOR

ADD FOREIGN KEY (ID\_PRODUCTO) REFERENCES PRODUCTO (ID\_PRODUCTO);

## Inserción de Datos

INSERT INTO PRODUCTO (ID\_PRODUCTO,CANTIDAD,ID\_DNI\_PROVEEDOR,CATEGORIA,SUBCATEGORIA,

ESTADO)

VALUES

(1,1,33786512,'Parte Arriba','Remera','Nuevo'),

(2,2,54207362,'Parte Abajo','Pantalon','Usado'),

(3,3,13022234,'Parte Abajo','Joggin', 'Nuevo'),

(4,4,16014361,'Parte Arriba','Sweater','Usado'),

(5,5,33786512,'Parte Arriba','Campera','Nuevo'),

(6,6,54207363,'Accesorios','Bufanda','Usado'),

(7,7,13022234,'Accesorios','Gorro','Nuevo'),

(8,4,16014361,'Accesorios','Cinturon','Usado'),

(9,9,35876512,'Parte Arriba','Camiseta','Nuevo'),

(10,18,32987123,'Calzado','Botas Lluvia','Usado'),

(11,4,16022876,'Calzado','Zapatilla Deportiva','Nuevo'),

(12,7,35876512,'Calzado','Sandalia','Usado'),

(13,5,32987123,'Parte Arriba','Camiseta','Nuevo'),

(14,12,16022876,'Parte Abajo','Pantalon de Vestir','Usado'),

(15,19,33786512,'Parte Abajo','Short','Usado');

INSERT INTO PROVEEDOR (ID\_PROVEEDOR,CBU,ID\_PRODUCTO,ID\_DNI\_PROVEEDOR)

VALUES (1,8176123266094573,1,33786512),

(2,9081763429675638,2,54207363),

(3,0000923156782395,3,13022234),

(4,1265342587692456,4,16014361),

(5,2736091275648972,5,33786512),

(6,8827465398126345,6,54207363),

(7,1212987125556339,7,13022234),

(8,5577822331983452,8,16014361),

(9,8823410956342584,9,35876512),

(10,118233350099276,10,32987123),

(11,3377441188735249,11,16022876),

(12,2298345672335544,12,35876512),

(13,3398765338762987,13,42987123),

(14,3998006651288734,14,16022876),

(15,1120098003356470,15,33786512);

INSERT INTO CLIENTES (ID\_DNI,DIRECCION,USUARIO,PROVINCIA)

VALUES

(33785612,'Alvarez Jonte 5090', 'Juan Perez', 'Buenos Aires'),

(54207362,'Conesa 4550', 'Elizabeth Taylor', 'Tucuman'),

(13022234,'Vieytes 486', 'Bob Marley', 'La Pampa'),

(16014361,'Adolfo Alsina 4489', 'Mauricio Macri', 'Cordoba'),

(23785612,'25 de Mayo 101', 'Florencia Bonelli', 'Rio Negro'),

(54207363,'Cosquin 1064', 'Daniel Bellizzi', 'Jujuy'),

(13012234,'Esmeralda 351', 'Wanda Nara', 'La Rioja'),

(16014363,'San Martin 451', 'Pedro Rodriguez', 'San Juan'),

(35875612,'Florida 450', 'Laura Fernandez', 'Mendoza'),

(32987123,'Mitre 128', 'Marta Roldan', 'San Luis'),

(16022876,'Agustín Alvarez 578', 'Monica Sanchez', 'Neuquen'),

(35876512,'Laprida 4712', 'Coqui Capitanich', 'Chaco'),

(32987133,'Brasil 987', 'Sofia Torres', 'Santa Fe'),

(16022877,'Peru 951', 'Ana Ramirez', 'Misiones'),

(98785612,'Mexico 967', 'Isabel Martinez', 'Corrientes');

INSERT INTOVENTAS(ID\_VENTA,ID\_PRODUCTO,CANTIDAD,FECHA\_DE\_VENTA,NUMERO\_FACTURA,ID\_DNI)

VALUES

(1,1,1,'2023-01-02',1,33785612),

(2,2,2,'2023-02-18',2,54207362),

(3,3,3,'2023-10-15',3,13022234),

(4,4,4,'2023-11-03',4,16014361),

(5,5,5,'2023-09-13',5,23785612),

(6,6,6,'2023-02-18',6,54207363),

(7,7,7,'2023-10-15',7,13012234),

(8,8,8,'2023-06-04',8,16014363),

(9,9,9,'2023-08-01',9,35875612),

(10,10,10,'2023-09-13',10,32987123),

(11,11,11,'2023-02-18',11,16022876),

(12,12,12,'2023-10-15',12,35876512),

(13,13,13,'2023-06-09',13,32987133),

(14,14,14,'2023-08-05',14,16022877),

(15,15,15,'2023-09-23',15,98785612);

INSERT INTO COBRANZAS (ID\_COBRANZAS,MONTO,FECHA\_DE\_COBRO,MEDIO\_PAGO,ID\_VENTA)

VALUES (1,58.7,'2023-05-05','Tarjeta',1),

(2,73.9,'2023-01-02','Efectivo',2),

(3,50.5,'2023-12-08','Efectivo',3),

(4,67.3,'2023-08-08','Tarjeta',4),

(5,23.6,'2023-12-09','Tarjeta',5),

(6,18.7,'2023-01-02','Tarjeta',6),

(7,91.6,'2023-02-11','Efectivo',7),

(8,15.7,'2023-03-01','Tarjeta',8),

(9,97.8,'2023-04-22','Efectivo',9),

(10,34.6,'2023-06-29','Tarjeta',10),

(11,88.1,'2023-07-09','Efectivo',11),

(12,88.1,'2023-08-05','Tarjeta',12),

(13,88.1,'2023-09-06','Efectivo',13),

(14,88.1,'2023-07-29','Efectivo',14),

(15,88.1,'2023-12-31','Tarjeta',15);

INSERT INTO COMPRAS (ID\_COMPRA,CANTIDAD,MONTO,ID\_PRODUCTO,ID\_PROVEEDOR,FECHA\_COMPRA)

VALUES (1,1,78.8,1,1,'2023-01-02'),

(2,2,90.1,1,1,'2023-02-18'),

(3,3,95.9,1,1,'2023-10-15'),

(4,4,33.8,1,1,'2023-11-03'),

(5,5,21.3,1,1,'2023-09-13'),

(6,6,90.1,1,1,'2023-02-18'),

(7,7,95.9,1,1,'2023-10-15'),

(8,8,33.8,1,1,'2023-06-04'),

(9,9,89.7,1,1,'2023-08-01'),

(10,10,21.3,1,1,'2023-09-13'),

(11,11,90.1,1,1,'2023-02-18'),

(12,12,95.9,1,1,'2023-10-15'),

(13,13,33.8,1,1,'2023-06-09'),

(14,14,89.7,1,1,'2023-08-05'),

(15,15,18.4,1,1,'2023-09-23');

INSERT INTO PRECIO\_COMPRAS (ID\_PRECIO\_COMPRA,PRECIO,FECHA\_COMPRA,ID\_PRODUCTO)

VALUES

(1,78.8,'2023-01-02',1),

(2,90.1,'2023-02-18',2),

(3,95.9,'2023-10-15',3),

(4,33.8,'2023-11-03',4),

(5,21.3,'2023-09-13',5),

(6,90.1,'2023-02-18',6),

(7,95.9,'2023-10-15',7),

(8,33.8,'2023-06-04',8),

(9,89.7,'2023-08-01',9),

(10,21.3,'2023-09-13',10),

(11,90.1,'2023-02-18',11),

(12,95.9,'2023-10-15',12),

(13,33.8,'2023-06-09',13),

(14,89.7,'2023-08-05',14),

(15,18.4,'2023-09-23',15);

INSERT INTO PRECIO\_VENTA (ID\_PRECIO\_VENTA,PRECIO,ID\_PRODUCTO,FECHA\_VENTA)

VALUES (1,85.3,1,'2023-01-02'),

(2,95.1,2,'2023-02-18'),

(3,96.9,3,'2023-10-15'),

(4,39.8,4,'2023-11-03'),

(5,29.3,5,'2023-09-13'),

(6,99.1,6,'2023-02-18'),

(7,99.9,7,'2023-10-15'),

(8,37.8,8,'2023-06-04'),

(9,96.7,9,'2023-08-01'),

(10,29.3,10,'2023-09-13'),

(11,99.1,11,'2023-02-18'),

(12,99.9,12,'2023-10-15'),

(13,39.8,13,'2023-06-09'),

(14,97.7,14,'2023-08-05'),

(15,19.4,15,'2023-09-23');

INSERT INTO PAGOS (ID\_PAGOS,ID\_COMPRA,MONTO,ID\_PROVEEDOR,FECHA\_DE\_PAGO,MEDIO\_PAGO)

VALUES (1,1,78.8,1,'2023-01-02','Efectivo'),

(2,2,90.1,2,'2023-02-18','Tarjeta'),

(3,3,95.9,3,'2023-10-15','Efectivo'),

(4,4,33.8,4,'2023-11-03','Efectivo'),

(5,5,21.3,5,'2023-09-13','Tarjeta'),

(6,6,90.1,6,'2023-02-18','Tarjeta'),

(7,7,95.9,7,'2023-10-15','Efectivo'),

(8,8,33.8,8,'2023-06-04','Efectivo'),

(9,9,89.7,9,'2023-08-01','Tarjeta'),

(10,10,21.3,10,'2023-09-13','Tarjeta'),

(11,11,90.1,11,'2023-02-18','Tarjeta'),

(12,12,95.9,12,'2023-10-15','Efectivo'),

(13,13,33.8,13,'2023-06-09','Efectivo'),

(14,14,89.7,14,'2023-08-05','Tarjeta'),

(15,15,18.4,15,'2023-09-23','Tarjeta');

INSERT INTO GASTOS

(ID\_GASTO,MONTO\_VEP,MONTO\_CONTADOR,MONTO\_FACTURA\_ML,AÑO\_MES,MONTO\_COMISION\_ML)

VALUES (1,90.5,99.5,17.8,'2023-01-01',15.9),

(2,90.5,99.5,17.8,'2023-02-01',15.9),

(3,90.5,99.5,17.8,'2023-03-01',15.9),

(4,90.5,99.5,17.8,'2023-04-01',15.9),

(5,90.5,99.5,17.8,'2023-05-01',15.9),

(6,90.5,99.5,17.8,'2023-06-01',15.9),

(7,90.5,99.5,17.8,'2023-07-01',15.9),

(8,90.5,99.5,17.8,'2023-08-01',15.9),

(9,90.5,99.5,17.8,'2023-09-01',15.9),

(10,90.5,99.5,17.8,'2023-10-01',15.9),

(11,90.5,99.5,17.8,'2023-11-01',15.9),

(12,90.5,99.5,17.8,'2023-12-01',15.9),

(13,90.5,99.5,17.8,'2023-01-01',15.9),

(14,90.5,99.5,17.8,'2023-02-01',15.9),

(15,90.5,99.5,17.8,'2023-03-01',15.9);

INSERT INTO IMPUESTOS (ID\_IMPUESTO,ID\_GASTO,PROVINCIA,MONTO,AÑO\_MES)

VALUES (1,1,'Mendoza',0.22,'2023-01-02'),

(2,2,'Neuquen',0.23,'2023-01-02'),

(3,3,'Mendoza',0.24,'2023-01-02'),

(4,4,'San Luis',0.29,'2023-01-02'),

(5,5,'Jujuy',0.21,'2023-01-02'),

(6,6,'Entre Rios',0.22,'2023-01-02'),

(7,7,'Salta',0.30,'2023-01-02'),

(8,8,'Buenos Aires',0.31,'2023-01-02'),

(9,9,'Santa Cruz',0.32,'2023-01-02'),

(10,10,'La Pampa',0.33,'2023-01-02'),

(11,11,'La Pampa',0.34,'2023-01-02'),

(12,12,'La Rioja',0.35,'2023-01-02'),

(13,13,'Formosa',0.36,'2023-01-02'),

(14,14,'Entre Rios',0.37,'2023-01-02'),

(15,15,'Mendoza',0.38,'2023-01-02');

## Vistas

Esta vista muestra el impuesto pagado más alto por Provincia, Año y mes:

CREATE VIEW vista\_impuesto\_mas\_alto AS

SELECT PROVINCIA, AÑO\_MES,

MAX(MONTO) AS impuesto\_mas\_alto

FROM IMPUESTOS

GROUP BY PROVINCIA,AÑO\_MES;

Esta vista muestra la cantidad de productos vendidos por estado, nuevo o usado:

CREATE VIEW Cantidad\_por\_estado AS

SELECT ESTADO, COUNT(\*) AS CANTIDAD

FROM PRODUCTO

GROUP BY ESTADO;

Esta vista muestra los cinco productos más vendidos por categoría:

CREATE VIEW vista\_productos\_mas\_vendidos AS

SELECT CATEGORIA, COUNT(\*) AS cantidad\_vendida

FROM PRODUCTO

JOIN VENTAS ON PRODUCTO.ID\_PRODUCTO = VENTAS.ID\_PRODUCTO

GROUP BY CATEGORIA

ORDER BY COUNT(\*) DESC

LIMIT 5;

Funciones:

Esta función muestra el total gastado por cliente

DELIMITER //

CREATE FUNCTION FN\_TOTAL\_GASTADO\_POR\_CLIENTE (ID\_CLIENTE INT) RETURNS FLOAT

BEGIN

DECLARE TOTAL\_GASTADO FLOAT;

SELECT SUM(MONTO) INTO TOTAL\_GASTADO

FROM COBRANZAS

WHERE ID\_VENTA IN (SELECT ID\_VENTA FROM VENTAS WHERE ID\_DNI = ID\_CLIENTE);

RETURN TOTAL\_GASTADO;

END //

DELIMITER ;

DELIMITER //

Esta función muestra los precios promedio de venta:

CREATE FUNCTION FN\_PRECIO\_PROMEDIO\_DE\_VENTA (ANO\_MES DATE) RETURNS FLOAT

BEGIN

DECLARE PROMEDIO FLOAT;

SELECT AVG(PRECIO) INTO PROMEDIO

FROM PRECIO\_VENTA

WHERE YEAR (FECHA\_VENTA) = YEAR (ANO\_MES) AND MONTH (FECHA\_VENTA) = MONTH (ANO\_MES);

RETURN PROMEDIO;

END //

DELIMITER ;

Procedimientos:

Este procedimiento ordena los clientes por provincia:

DELIMITER //

CREATE PROCEDURE SP\_OrdenarTabla\_Clientes (

IN CampoDeOrdenamiento VARCHAR(50), -- Parámetro para el campo de ordenamiento

IN Orden BOOLEAN) -- Parámetro para el orden (ascendente o descendente

BEGIN

SET @query = CONCAT('SELECT ID\_DNI, USUARIO, PROVINCIA FROM clientes ORDER BY ' , CampoDeOrdenamiento , IF(Orden, ' ASC', ' DESC'));

PREPARE consulta FROM @query;

EXECUTE consulta;

DEALLOCATE PREPARE consulta;

END //

DELIMITER ;

---PARA EJECUTARLO---

CALL SP\_OrdenarTabla\_Clientes ('PROVINCIA', 1);

CALL SP\_OrdenarTabla\_Clientes ('PROVINCIA', 0);

El siguiente procedimiento inserta un registro de cliente:

DELIMITER //

CREATE PROCEDURE SP\_Insertar\_Cliente (

IN DNI INT(11),

IN DIRECCION VARCHAR(50),

IN NOMBRE\_APELLIDO VARCHAR(50),

IN PROVINCIA VARCHAR (50))

BEGIN

SET @query = CONCAT('INSERT INTO feria\_placarcito.clientes (ID\_DNI, DIRECCION, USUARIO, PROVINCIA) VALUES (', DNI, ', "', DIRECCION, '", "', NOMBRE\_APELLIDO, '", "', PROVINCIA, '")');

PREPARE consulta FROM @query;

EXECUTE consulta;

DEALLOCATE PREPARE consulta;

END //

DELIMITER ;

-----Para ejecutarlo----

CALL SP\_Insertar\_Cliente (21424069, 'Mitre 481', 'Pepito Boton', 'Buenos Aires');

SELECT \* FROM CLIENTES;

Triggers:

Creación de Tabla LOG\_CLIENTE y LOG\_PRODUCTO:

DROP TABLE IF EXISTS LOG\_CLIENTES;

CREATE TABLE IF NOT EXISTS LOG\_CLIENTES (

ID\_DNI INT AUTO\_INCREMENT PRIMARY KEY,

ID\_CLIENTE INT NOT NULL,

ACTION\_TYPE VARCHAR(50) NOT NULL,

ACTION\_DATE DATE NOT NULL,

CLIENTE\_ACCION VARCHAR(255) NOT NULL,

FOREIGN KEY (ID\_DNI) REFERENCES CLIENTES (ID\_DNI)

);

DROP TABLE IF EXISTS LOG\_PRODUCTO;

CREATE TABLE IF NOT EXISTS LOG\_PRODUCTO (

ID\_PRODUCTO INT AUTO\_INCREMENT PRIMARY KEY,

ID\_PRODUCTO\_ORIGINAL INT NOT NULL,

ACTION\_TYPE VARCHAR(50) NOT NULL,

ACTION\_DATE DATE NOT NULL,

ACTION\_TIME TIME NOT NULL,

PRODUCTO\_ACCION VARCHAR (255) NOT NULL,

FOREIGN KEY (ID\_PRODUCTO\_ORIGINAL) REFERENCES PRODUCTO (ID\_PRODUCTO)

);

**TRIGGER 1:** Registrar inserciones de nuevos clientes

DROP TRIGGER IF EXISTS CLIENTE\_BEFORE\_INSERT;

DELIMITER //

CREATE TRIGGER CLIENTE\_BEFORE\_INSERT

BEFORE

INSERT

ON CLIENTES FOR EACH ROW BEGIN

INSERT INTO LOG\_CLIENTES (

ID\_CLIENTE,

ACTION\_TYPE,

ACTION\_DATE,

ACTION\_TIME,

CLIENTE\_ACCION

)

VALUES

(NEW\_ID\_CLIENTE,

'INSERT',

CURDATE(),

CURTIME(),

CONCAT('Nuevo Cliente creado: ', new.ID\_DNI)

);

END //

DELIMITER ;

**TRIGGER 2:** Registrar actualizaciones de productos

DROP TRIGGER IF EXISTS PRODUCTO\_AFTER\_UPDATE;

DELIMITER //

CREATE TRIGGER PRODUCTO\_AFTER\_UPDATE

AFTER

UPDATE

ON PRODUCTO FOR EACH ROW BEGIN

INSERT INTO LOG\_PRODUCTO (

ID\_PRODUCTO\_ORIGINAL,

ACTION\_TYPE,

ACTION\_DATE,

ACTION\_TIME,

PRODUCTO\_ACCION

)

VALUES (

NEW.ID\_PRODUCTO,

'UPDATE',

CURDATE(),

CURTIME(),

CONCAT('Producto Actualizado: ' , NEW.ID\_PRODUCTO)

);

END //

DELIMITER ;