

Proyecto SQL – Feria Placarcito

Profesor: REDONDO, Camilo Andrés

Tutor: LOPEZ SAEZ, José Ignacio

Alumna: BELLIZZI, Florencia Belén

Comisión: N° 47365

DNI: 33.786.512

Link: <https://placarcito.mercadoshops.com.ar/>

Link Github: <https://github.com/Dankex-Bell/SQL.git>

# Introducción

## Descripción de la Situación de Negocio

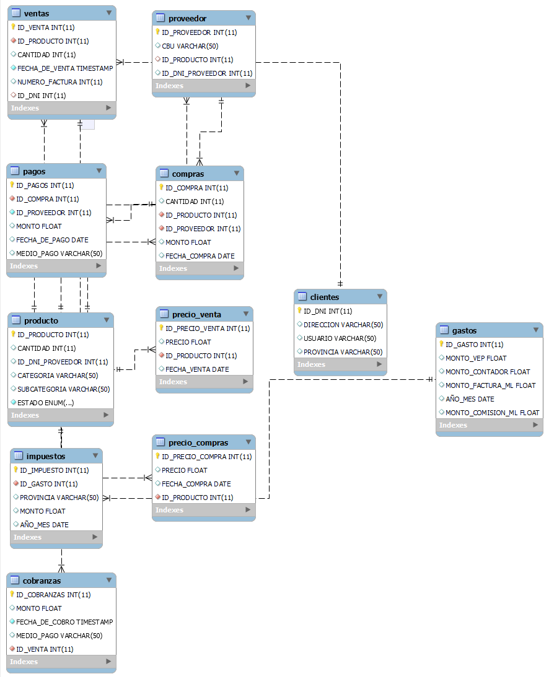
En la feria placarcito, nos dedicamos a facilitar la venta de productos nuevos y usados en una variedad de categorías, como ropa, libros, juguetes, carteras y zapatos. Además, ofrecemos un excelente servicio post venta brindándole al usuario la posibilidad de devolver el producto dentro de los 30 días de entregado, sea por el motivo que decida. Contamos con excelentes 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.

# Entidades y relaciones

## Diagrama Entidad-Relación



## Descripción de tablas

### Tabla: Producto

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

|  |  |  |
| --- | --- | --- |
| **CAMPOS** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_PRODUCTO | NUMERICO | Identificación del Producto |
| CANTIDAD | NUMERICO | Cantidad Vendida |
| ID\_DNI\_PROVEEDOR | NUMERICO | ID de DNI |
| CATEGORIA | TEXTO | Categoría de Prenda |
| SUBCATEGORIA | TEXTO | Subcategoría de Prenda |
| ESTADO | ENUM | Nuevo o Usado |

### Tabla: Proveedor

Esta tabla almacena información de los proveedores de placarcito.

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_PROVEEDOR | NUMERICO | Identificación del Proveedor |
| CBU /ALIAS | TEXTO | Dato de pago al proveedor |
| ID\_PRODUCTO | NUMERICO | Identificación de producto |
| ID\_DNI\_PROVEEDOR | NUMERICO | Identificación DNI del proveedor |

### 

### Tabla: Clientes

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_DNI | NUMERICO | Identificación DNI del Cliente |
| DIRECCION | TEXTO | Residencia del Cliente |
| USUARIO | TEXTO | Usuario utilizado |
| PROVINCIA | TEXTO | Provincia de residencia |

### Tabla: Cobranzas

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_COBRANZAS | NUMERICO | Identificación de Cobranza |
| MONTO | NUMERICO | Valor |
| FECHA\_DE\_COBRO | FECHA | Fecha de cobro |
| MEDIO\_PAGO | TEXTO | Medio para realizar el pago |
| ID\_VENTA | NUMERICO | Identificación de Venta |

### Tabla: Pagos

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_PAGOS | NUMERICO | Identificación de Pago realizado al proveedor |
| ID\_COMPRA | NUMERICO | Identificación de Compra |
| MONTO | NUMERICO | Valor |
| ID\_PROVEEDOR | NUMERICO | Identificación de Proveedor |
| FECHA\_DE\_PAGO | FECHA | Fecha de pago |
| MEDIO\_DE\_PAGO | TEXTO | Medio para realizar el pago |

### Tabla: Compras

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_COMPRA | NUMERICO | Identificación de Compra |
| CANTIDAD | NUMERICO | Cantidad Comprada |
| MONTO | NUMERICO | Valor |
| ID\_PRODUCTO | NUMERICO | Identificación de Producto |
| ID\_PROVEEDOR | NUMERICO | Identificación de Proveedor |
| FECHA\_DE\_COMPRA | FECHA | Fecha de compra del producto |

### Tabla: Precio de Compras

Esta tabla contiene información de los precios de cada producto comprado a los proveedores.

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_PRECIO\_COMPRA | NUMERICO | Identificación de Precio de Compra |
| PRECIO | NUMERICO | Valor |
| FECHA\_COMPRA | FECHA | Fecha de la compra al proveedor |
| ID\_PRODUCTO | NUMERICO | Identificación del Producto |

### Tabla: Precio de Ventas

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_PRECIO\_VENTA | NUMERICO | Identificación de Precio de Venta |
| PRECIO | NUMERICO | Valor |
| ID\_PRODUCTO | NUMERICO | Identificación del Producto vendido |
| FECHA\_VENTA | FECHA | Fecha de la venta en ML |

### Tabla: Gastos

Esta tabla registra los gastos mensuales que genera la feria.

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_GASTO | NUMERICO | Identificación del Gasto |
| MONTO\_VEP | NUMERICO | Valor mensual a pagar |
| MONTO\_CONTADOR | NUMERICO | Valor mensual a pagar al contador |
| MONTO\_FACTURA\_ML | NUMERICO | Valor a pagar mensualmente a ML |
| AÑO\_MES | FECHA | Año y Mes del gasto |
| MONTO\_COMISION\_ML | NUMERICO | Monto mensual de comisión de ML |

### Tabla: Ventas

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_VENTA | NUMERICO | Identificación de la venta |
| ID\_PRODUCTO | NUMERICO | Identificación del producto vendido |
| CANTIDAD | NUMERICO | Cantidad vendida |
| FECHA\_VENTA | FECHA | Fecha de la venta |
| NUMERO\_FACTURA | NUMERICO | Numero de factura en AFIP |
| ID\_DNI | NUMERICO | Identificación del DNI |

### Tabla: Impuestos

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

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **TIPO DE DATO** | **DESCRIPCION** |
| ID\_IMPUESTO | NUMERICO | Identificación del Impuesto |
| ID\_GASTO | NUMERICO | Identificación del Gasto |
| PROVINCIA | TEXTO | Provincia Argentina |
| MONTO | VALOR | Valor a pagar |
| AÑO\_MES | FECHA | Fecha de pago del impuesto |

## Inserción de Datos

La Inserción de Datos se realizó de forma manual y es ficticia. Se insertaron 15 registros en cada tabla.

## 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:

**La siguiente función muestra el total gastado por cliente.**

**Es un dato importante ya que por el monto gastado solemos enviar regalos para captar al cliente y que realice otra compra ya no desde Mercado Libre sino directamente desde el link de la feria en Mercado Shops, donde la comisión a pagar es mucho menor por lo tanto nos da un margen de ganancia mucho más amplio :**

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;

**Esta función muestra los precios promedio de venta:**

**Esta función es sumamente importante ya que con la inflación que hay en Argentina, es importante saber si el promedio de ticket de venta es bajo y hay que actualizarlo.**

DELIMITER //  
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 tomando como dato de entrada cualquier provincia argentina, y de salida el número 1 para orden ascendente y 0 para orden descendente. Es de mucha utilidad para saber en qué provincia se vende más:**

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 ;

**El siguiente procedimiento inserta un registro de cliente nuevo:**

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 ;

Triggers:

**Se crearon dos tablas de auditoría para registrar si se creó algún nuevo cliente y otra tabla de auditoría para registrar la actualización de algún producto.**

**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 ;

DCL- Creación de usuarios y permisos:

*/\*Creacion de usuario con permisos solo de lectura\*/*  
CREATE USER 'solo\_lectura'@'localhost'IDENTIFIED BY'contrasena\_solo\_lectura';  
  
*/\*Conceder permisos solo de lectura en todas las tablas de Placarcito\*/*  
*/\*No se permite eliminar registros\*/*  
GRANT SELECT ON feria\_placarcito.\* TO 'solo\_lectura'@'localhost';  
  
*/\*Creacion de usuario lectura\_insercion\_modificacion con permisos de*

*lectura, insercion y modificación\*/*  
CREATE USER 'lectura\_insercion\_modificacion'@'localhost' IDENTIFIED BY  
'contrasena\_lectura\_insercion\_modificacion';  
  
*/\*Conceder permisos  de lectura, insercion y modificacion en todas las tablas de placarcito\*/*  
*/\*No se permite eliminar registros\*/*  
GRANT SELECT, INSERT, UPDATE ON feria\_placarcito.\* TO  
'lectura\_insercion\_modificacion'@'localhost';  
  
*/\*Eliminar la capacidad de eliminar registros para ambos usuarios\*/*  
REVOKE DELETE ON \*.\* FROM 'solo\_lectura'@'localhost';  
  
REVOKE DELETE ON \*.\* FROM 'lectura\_insercion\_modificacion'@'localhost';

TCL -Transacciones

*/\*Iniciar una transacción en la tabla 'producto'\*/*  
START TRANSACTION;  
*/\*Eliminar algunos registros en la tabla 'producto' (o reemplazar por insercion si esta vacía)\*/*  
*/\*Comentar la sentencia Rollback (no ejecutar)\*/*  
*/\*ROLLBACK;*  
*/\*Comentar la sentencia Commit (no ejecutar)\*/*  
*/\*COMMIT;*  
*/\*Iniciar una transaccion en la tabla 'proveedor'\*/*

START TRANSACTION;  
*/\*Insertar ocho nuevos registros de productos en la tabla 'producto'\*/*

INSERT INTO producto  
            (  
                        id\_producto,  
                        cantidad,  
                        id\_dni\_proveedor,  
                        categoria,  
                        subcategoria,  
                        estado  
            )  
            VALUES  
            (  
                        16,20,33786512,  
                        'Parte Abajo',  
                        'Jean',  
                        'Usado'  
            )  
            ,  
            (  
                        17,21,33786512,  
                        'Parte Abajo',  
                        'Pantalon de Vestir',  
                        'Usado'  
            )  
            ,  
            (  
                        18,26,33786512,  
                        'Parte Abajo',  
                        'Jean',  
                        'Usado'  
            )  
            ,  
            (  
                        19,29,33786512,  
                        'Parte Abajo',  
                        'Short',  
                        'Nuevo'  
            );  
  
*/\*Agregar un savepoint después de la insercion del cuarto registro\*/*

SAVEPOINT savepoint4;  
*/\*Continuar insertando registros\*/*

INSERT INTO producto  
            (  
                        id\_producto,  
                        cantidad,  
                        id\_dni\_proveedor,  
                        categoria,  
                        subcategoria,  
                        estado  
            )  
            VALUES  
            (  
                        20,20,33786512,  
                        'Parte Abajo',  
                        'Jean',  
                        'Nuevo'  
            )  
            ,  
            (  
                        21,51,33786512,  
                        'Parte Abajo',  
                        'Pantalon de Vestir',  
                        'Usado'  
            )  
            ,  
            (  
                        22,66,33786512,  
                        'Parte Abajo',  
                        'Jean',  
                        'Nuevo'  
            )  
            ,  
            (  
                        23,39,33786512,  
                        'Parte Abajo',  
                        'Short',  
                        'Nuevo'  
            );  
  
*/\*Agregar un savepoint despues de la insercion del octavo registro\*/*

SAVEPOINT savepoint8;  
*/\*Comentar la sentencia para eliminar el savepoint de los primeros 4 registros insertados\*/*  
*/\*ROLLBACK TO savepoint4;*

## Conclusiones

La creación de la base de datos permitirá:

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, 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 5000

Tecnologías Utilizadas:

* Se utilizó la siguiente página para los scripts: <https://www.dpriver.com/pp/sqlformat.htm>
* MySQL Workbench
* Para la elaboración del Diagrama Entidad relación: <https://erdplus.com/standalone>