Sprint5 Grupo Verde  
  
Cecilia Montero  
Karla Mieres  
Zimram Blanco  
Gonzalo Aranda  
Andrea Pilquiman  
Felipe Parra  
  
SPRINT DE ENTREGA:

Se solicita como entregable de este Sprint la implementación final de todos los conceptos vistos durante el Módulo 2: Bases de datos. Por tanto, se debe poner foco en lo siguiente:

Comentar debidamente el código para que sea comprensible por un tercero.

El script SQL debe utilizarse para crear la estructura de la base de datos, realizar operaciones en la base de datos (p. ej. rellenarla con datos) y cambiar o eliminar la estructura de la base de datos.

Deben crear un usuario con privilegios para crear, eliminar y modificar tablas, insertar registros.

Nuestra tienda virtual ha crecido mucho estas últimas semanas. No solo aumentó significativamente el número de colaboradores y usuarios, sino que también los productos disponibles. En este sentido, nos pidieron que diseñemos una base de datos capaz de satisfacer la creciente demanda de información y datos.

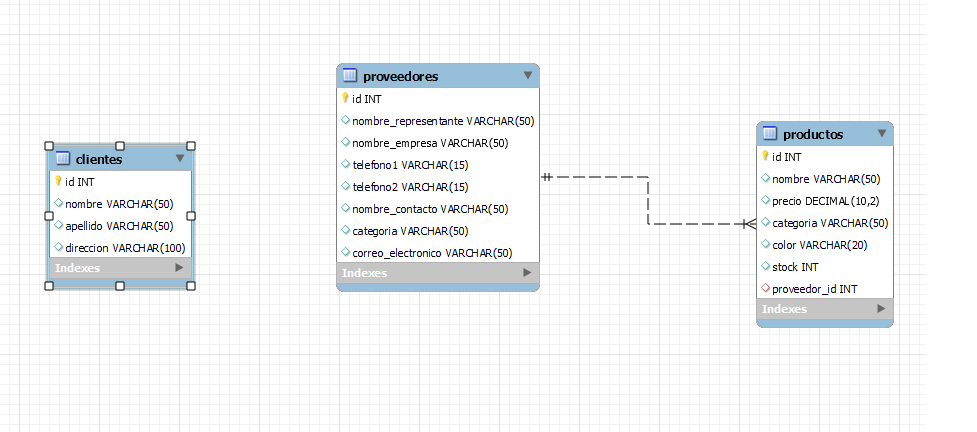
TeLoVendo recibe productos de diferentes proveedores para comercializarlos. Cada proveedor debe informarnos el nombre del representante legal, su nombre corporativo, al menos dos teléfonos de contacto (y el nombre de quien recibe las llamadas), la categoría de sus productos (solo nos pueden indicar una categoría) y un correo electrónico para enviar la factura. Sabemos que la **mayoría** de los proveedores son de productos electrónicos.

Agregue 5 proveedores a la base de datos. En general, los proveedores venden **muchos** productos.

TeLoVendo **tiene actualmente muchos** clientes, pero nos piden que ingresemos solo 5 para probar la nueva base de datos. Cada cliente tiene un nombre, apellido, dirección (solo pueden ingresar una).

TeLoVendo tiene diferentes productos. Ingrese 10 productos y su respectivo stock. Cada producto tiene información sobre su precio, su categoría, proveedor y color. Los productos pueden tener **muchos** proveedores.

Como un entregable, nos piden que diseñemos un diagrama entidad relación sólo con la información que tenemos. En caso de tener nuevas ideas respecto a futura información requerida y nuevas entidades, solo nos piden que la indiquemos en un archivo .docx.



A partir del diagrama, debemos construir un script que cree tablas de acuerdo a las entidades e ingrese datos.

- Cuál es la categoría de productos que **más se repite**.

- Cuáles son los productos con **mayor stock**

- Qué color de **producto es más común** en nuestra tienda.

- Cuál o cuáles son los **proveedores con menor stock de productos**.

Por último:

- Cambien la categoría de productos **más popular** por ‘Electrónica y computación’.

Deben subir el trabajo a un repositorio en Git-Hub.

A modo de entrega, se debe disponer un documento Word o PDF en el que se indique:

- Los nombres de los integrantes del equipo

- Ruta del repositorio en GitHub

Consideraciones adicionales

- El código **debe estar debidamente comentado**

create schema telovendo\_sprint;

CREATE USER 'equipo\_verde'@'localhost' IDENTIFIED BY 'verde';

GRANT CREATE, DROP, ALTER, INSERT ON \*.\* TO 'equipo\_verde'@'localhost';

FLUSH PRIVILEGES;

USE telovendo\_sprint;

-- Creamos tabla de proveedores

CREATE TABLE

proveedores (

id INT PRIMARY KEY,

nombre\_representante VARCHAR(50),

nombre\_corporativo VARCHAR(50),

telefono1 VARCHAR(15),

telefono2 VARCHAR(15),

nombre\_empresa VARCHAR(50),

nombre\_contacto VARCHAR(50),

categoria VARCHAR(50),

correo\_electronico VARCHAR(50)

);

-- Creamos tabla de clientes

CREATE TABLE

clientes (

id INT PRIMARY KEY,

nombre VARCHAR(50),

apellido VARCHAR(50),

direccion VARCHAR(100)

);

-- Creamos tabla de productos

CREATE TABLE

productos (

id INT PRIMARY KEY,

nombre VARCHAR(50),

precio DECIMAL(10, 2),

categoria VARCHAR(50),

color VARCHAR(20),

stock INT,

proveedor\_id INT,

FOREIGN KEY (proveedor\_id) REFERENCES proveedores(id)

);

-- Insertar datos de proveedores

INSERT INTO proveedores (id, nombre\_representante, nombre\_empresa, telefono1, telefono2, nombre\_contacto, categoria, correo\_electronico)

VALUES

(1, 'Miguel Ángel Gómez', 'TechSoluciones', '1111111111', '2222222222', 'Juan Carlos López', 'Electrónicos', 'miguel@example.com'),

(2, 'Gabriel García Márquez', 'TechInnovación', '3333333333', '4444444444', 'Luisa Fernández', 'Electrónicos', 'gabriel@example.com'),

(3, 'Fernanda Silva', 'TechProyectos', '5555555555', '6666666666', 'María Rodríguez', 'Electrónicos', 'fernanda@example.com'),

(4, 'Sebastián Hernández', 'TechSoluciones', '7777777777', '8888888888', 'Sofía Gutiérrez', 'Electrónicos', 'sebastian@example.com'),

(5, 'Valentina González', 'TechInnovación', '9999999999', '0000000000', 'Camila López', 'Electrónicos', 'valentina@example.com');

Select \* from proveedores;

-- Insertar datos de clientes

INSERT INTO clientes (id, nombre, apellido, direccion)

VALUES

(1, 'Miguel', 'Gómez', 'Calle Barros Arana 123'),

(2, 'Gabriel', 'Márquez', 'Avenida OHiggins 456'),

(3, 'Fernanda', 'Silva', 'Calle Aníbal Pinto 789'),

(4, 'Sebastián', 'Hernández', 'Avenida Los Carrera 987'),

(5, 'Valentina', 'González', 'Calle Freire 654');

Select \* from clientes;

-- Insertar datos de productos electrónicos

INSERT INTO productos (id, nombre, precio, categoria, proveedor\_id, color, stock)

VALUES

(1, 'iPhone 13 Pro', 129990, 'Smartphones', 1, 'Gris', 50),

(2, 'Samsung Galaxy S21 Ultra', 119900, 'Smartphones', 2, 'Negro', 40),

(3, 'Sony WH-1000XM4', 349000, 'Audífonos', 2, 'Negro', 30),

(4, 'LG C1 OLED TV', 199900, 'Televisores', 1, 'Negro', 20),

(5, 'DJI Mavic Air 2', 79900, 'Drones', 5, 'Gris', 10),

(6, 'Nintendo Switch', 29900, 'Consolas de videojuegos', 5, 'Rojo/Negro', 15),

(7, 'Sony PlayStation 5', 499000, 'Consolas de videojuegos', 3, 'Blanco', 5),

(8, 'Apple MacBook Pro', 1999000, 'Computadoras portátiles', 4, 'Gris', 25),

(9, 'Samsung Q90T QLED TV', 179900, 'Televisores', 4, 'Negro', 15),

(10, 'GoPro HERO9 Black', 44900, 'Cámaras de acción', 3, 'Negro', 8);

select \* from productos;

-- a) Categoría de productos que más se repite:

SELECT categoria, COUNT(\*) AS cantidad

FROM productos

GROUP BY categoria

ORDER BY cantidad DESC

LIMIT 1;

-- b) Los 5 Productos con mayor stock

SELECT \* FROM productos ORDER BY stock DESC LIMIT 5;

-- c) Color de producto más común

SELECT color, COUNT(\*) AS cantidad

FROM productos

GROUP BY color

ORDER BY cantidad DESC

LIMIT 1;

-- d) Proveedor(es) con menor stock de productos

SELECT proveedores.\* FROM proveedores

INNER JOIN productos ON proveedores.id = productos.proveedor\_id

GROUP BY proveedores.id

ORDER BY

SUM(productos.stock) ASC

LIMIT 1;

-- Desactivamos el modo de actualización segura (safe update mode)

SET SQL\_SAFE\_UPDATES = 0;

-- Cambiar la categoría de productos más popular por 'Electrónica y computación':

UPDATE productos

SET categoria = 'Electrónica y computación'

WHERE categoria = (

SELECT categoria

FROM (

SELECT categoria, COUNT(\*) AS count

FROM productos

GROUP BY categoria

ORDER BY count DESC

LIMIT 1

) AS subquery

);

-- Activamos el modo de actualización segura (safe update mode)

SET SQL\_SAFE\_UPDATES = 1;