**PARCIAL DE BASE DE DATOS**

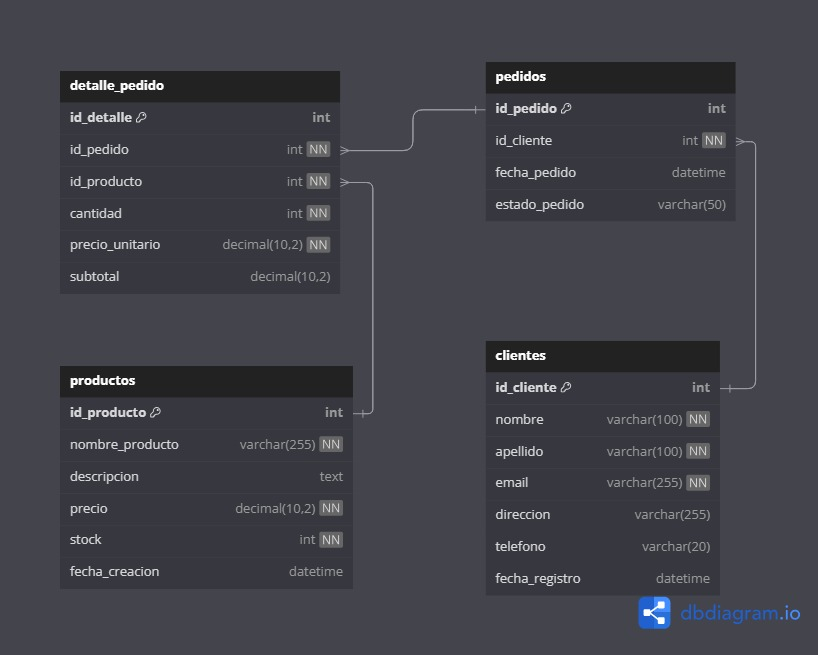
|  |
| --- |
| Apellido y Nombre: Finelli Teo Marco, Borgiattino Maximo, Banega Héctor y Marquez Lucas |
| División: I-TSIA.-S - 1 / 1 NMS |
| Modalidad: Semi-Presencial |

Introducción

Con el grupo tomamos la decisión de generar una base de datos para la administración de una tienda online de Notebooks, ya que creemos que cumple perfectamente con el alcance y los requisitos del trabajo.

Al principio generamos tres tablas, una de clientes, una de pedidos y otra de productos. Sin embargo al pulir el trabajo nos topamos con que nuestro sistema solo permitia que el cliente hiciera el pedido de un solo producto en la cantidad deseada, caso no real ya que podemos comprar libremente la cantidad de productos que necesitemos.

A raíz de esta situación, optamos por implementar una cuarta tabla que permitiera detallar correctamente qué productos encargó el cliente y en qué cantidad.

A continuación, presentamos el diagrama entidad-relación para comprender el sistema.

**PARTE 1: DESARROLLO DEL SCRIPT COMPLETO**

**Estructura del Script Requerido**

**1. CREACIÓN DE BASE DE DATOS (DDL):**

**· Crear modelo de base de datos**

**· Implementar mínimo 3 tablas relacionadas con claves primarias y foráneas**

**· Aplicar comandos DDL para estructura completa**

**· Incluir índices para optimización**

Script:

CREATE DATABASE IF NOT EXISTS notebooks;

USE notebooks;

CREATE TABLE clientes (

id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

email VARCHAR(255) NOT NULL,

direccion TEXT,

telefono VARCHAR(20)

);

CREATE TABLE productos (

id\_producto INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_producto VARCHAR(255) NOT NULL,

descripcion TEXT,

precio DECIMAL(10,2) NOT NULL,

stock INT NOT NULL DEFAULT 0,

fecha\_creacion DATETIME DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE pedidos (

id\_pedido INT AUTO\_INCREMENT PRIMARY KEY,

id\_cliente INT NOT NULL,

fecha\_pedido DATETIME,

estado VARCHAR(50),

FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente)

);

CREATE TABLE detalle\_pedido (

id\_detalle INT AUTO\_INCREMENT PRIMARY KEY,

id\_pedido INT NOT NULL,

id\_producto INT NOT NULL,

cantidad INT NOT NULL,

total DECIMAL(10,2) NOT NULL,

FOREIGN KEY (id\_pedido) REFERENCES pedidos(id\_pedido),

FOREIGN KEY (id\_producto) REFERENCES productos(id\_producto)

);

**2. INSERCIÓN DE DATOS (DML):**

**· Insertar datos realistas y consistentes**

**· Mínimo 20 registros por tabla principal**

**· Datos que permitan análisis significativo**

INSERT INTO clientes (nombre, apellido, email, direccion, telefono) VALUES

('Lucas','García','lucas.garcía@mail.com','Pasaje Los Andes 977, Buenos Aires','+5495740690485'),

('Mateo','Martínez','mateo.martínez@correo.com','Calle Falsa 827, Mar del Plata','+5493176189436'),

('Sofía','Rodríguez','sofía.rodríguez@example.com','Ruta Nacional 169, Rosario','+5491158013068'),

('Valentina','Fernández','valentina.fernández@mail.com','Pasaje Los Andes 317, Córdoba','+5497530678790'),

('Benjamín','Pérez','benjamín.pérez@example.com','Bv. Central 166, Salta','+5491967814675'),

('Isabella','López','isabella.lópez@example.com','Calle Falsa 689, Mendoza','+5494635222954'),

('Joaquín','González','joaquín.gonzález@correo.com','Pasaje Los Andes 860, Tucumán','+5499330465614'),

('Catalina','Díaz','catalina.díaz@correo.com','Calle Falsa 108, Buenos Aires','+5491444765187'),

('Agustín','Sánchez','agustín.sánchez@mail.com','Ruta Nacional 678, Tucumán','+5493634497157'),

('Emilia','Romero','emilia.romero@example.com','Calle Falsa 612, Tucumán','+5492295905328'),

('Facundo','Torres','facundo.torres@example.com','Ruta Nacional 967, Mendoza','+5499815244865'),

('Julieta','Ruiz','julieta.ruiz@example.com','Av. Siempre Viva 246, Rosario','+5496728504969'),

('Santiago','Flores','santiago.flores@correo.com','Bv. Central 889, Salta','+5498899067544'),

('Olivia','Acosta','olivia.acosta@mail.com','Calle Falsa 918, Tucumán','+5498781780976'),

('Martina','Herrera','martina.herrera@example.com','Bv. Central 619, Buenos Aires','+5493300598997'),

('Lautaro','Benítez','lautaro.benítez@example.com','Ruta Nacional 645, Buenos Aires','+5491286946148'),

('Victoria','Ortiz','victoria.ortiz@example.com','Bv. Central 535, Buenos Aires','+5492171541487'),

('Manuel','Moreno','manuel.moreno@example.com','Ruta Nacional 894, Rosario','+5493892103725'),

('Abril','Giménez','abril.giménez@mail.com','Ruta Nacional 764, Córdoba','+5498772785818'),

('Thiago','Cabrera','thiago.cabrera@mail.com','Bv. Central 429, Tucumán','+5495098482036');

INSERT INTO productos (id\_producto, nombre\_producto, descripcion, precio, stock) VALUES

(1,'Notebook Lenovo IdeaPad 3 Intel Core i3 8GB 256GB SSD 15.6" FreeDOS', 'Notebook con buen rendimiento para uso diario y estudio. Procesador Intel Core i3.', 549999.00, 20),

(2,'Notebook HP 15-dw0000la Intel Core i5 8GB 256GB SSD 15.6" FreeDOS', 'Notebook versátil para trabajo y entretenimiento. Procesador Intel Core i5.', 619999.00, 15),

(3,'Notebook ASUS Vivobook 15 M1502YA AMD Ryzen 5 16GB 512GB SSD 15.6" FreeDOS', 'Ideal para multitarea y tareas exigentes. Procesador AMD Ryzen 5 y 16GB de RAM.', 699999.00, 10),

(4,'Notebook Dell Inspiron 3520 Intel Core i7 16GB 512GB SSD 15.6" FreeDOS', 'Alto rendimiento para profesionales y usuarios avanzados. Procesador Intel Core i7.', 849999.00, 8),

(5,'Notebook Acer Aspire 3 A315-59 Intel Core i5 8GB 512GB SSD 15.6" FreeDOS', 'Diseño práctico y eficiente para el día a día. Procesador Intel Core i5.', 649999.00, 18),

(6,'Notebook MSI Thin GF63 12UC-625AR Intel Core i7 8GB 512GB SSD NVIDIA RTX 3050 15.6" FreeDOS', 'Notebook gaming para experiencias inmersivas. Tarjeta gráfica NVIDIA RTX 3050.', 1199999.00, 5),

(7,'Notebook HP 15-fc0000la AMD Ryzen 3 8GB 256GB SSD 15.6" FreeDOS', 'Opción económica para tareas básicas y navegación. Procesador AMD Ryzen 3.', 499999.00, 25),

(8,'Notebook ASUS ROG Strix G15 G513RC AMD Ryzen 7 16GB 512GB SSD NVIDIA RTX 3050 15.6" FreeDOS', 'Potente notebook gaming con AMD Ryzen 7 y RTX 3050. Ideal para gamers.', 1399999.00, 3),

(9,'Notebook Lenovo IdeaPad 3 15IAU7 Intel Core i5 8GB 512GB SSD 15.6" FreeDOS', 'Notebook eficiente para trabajo y estudio. Procesador Intel Core i5 de última generación.', 699999.00, 12),

(10,'Notebook Acer Aspire 5 A515-57 Intel Core i7 16GB 512GB SSD 15.6" FreeDOS', 'Rendimiento superior para tareas exigentes. Intel Core i7 y 16GB de RAM.', 899999.00, 7),

(11,'Notebook HP Victus 15-fa1000la Intel Core i5 8GB 512GB SSD NVIDIA RTX 2050 15.6" FreeDOS', 'Notebook gaming de entrada con buen desempeño gráfico. Procesador Intel Core i5 y RTX 2050.', 999999.00, 9),

(12,'Notebook Dell Latitude 3520 Intel Core i3 8GB 256GB SSD 15.6" FreeDOS', 'Diseñada para productividad y fiabilidad en entornos empresariales. Intel Core i3.', 579999.00, 18),

(13,'Notebook ASUS TUF Gaming F15 FX506HC-HN004 Intel Core i5 8GB 512GB SSD NVIDIA RTX 3050 15.6" FreeDOS', 'Potente máquina para gaming y multitarea. Intel Core i5 y RTX 3050.', 1099999.00, 6),

(14,'Notebook EXO Smart E26 Core i3 8GB 240GB SSD 15.6" FreeDOS', 'Opción económica para uso diario y educativo. Procesador Intel Core i3.', 479999.00, 22),

(15,'Notebook Gigabyte G5 GE-51LA263SH Intel Core i5 16GB 512GB SSD NVIDIA RTX 3050 15.6" FreeDOS', 'Notebook gaming de alto rendimiento con gráficos dedicados y 16GB de RAM.', 1249999.00, 4),

(16,'Notebook HP 250 G9 Intel Core i7 16GB 512GB SSD 15.6" FreeDOS', 'Ideal para profesionales, ofrece rendimiento y fiabilidad. Intel Core i7.', 949999.00, 10),

(17,'Notebook MSI Katana 15 B13VEK-1090XAR Intel Core i7 16GB 1TB SSD NVIDIA RTX 4050 15.6" FreeDOS', 'Gama alta para gaming exigente y creación de contenido. Intel Core i7 y RTX 4050.', 1699999.00, 3),

(18,'Notebook Lenovo IdeaPad Slim 3 Intel Core i7 16GB 512GB SSD 15.6" FreeDOS', 'Diseño delgado y ligero con gran potencia para tareas avanzadas. Intel Core i7.', 879999.00, 11),

(19,'Notebook Lenovo V15 G2 ALC AMD Ryzen 5 8GB 256GB SSD 15.6" FreeDOS', 'Ideal para profesionales y estudiantes, con un procesador Ryzen 5 para buen rendimiento.', 639999.00, 14),

(20,'Notebook HP Pavilion 15-eg2000la Intel Core i7 16GB 512GB SSD 15.6" FreeDOS', 'Diseño elegante y potente para tareas creativas y multitarea intensiva. Core i7.', 999999.00, 9);

INSERT INTO pedidos (id\_pedido, id\_cliente, fecha\_pedido, estado) VALUES

(1,2,'2025-06-01 00:00:00','enviado'),

(2,13,'2025-06-02 00:00:00','entregado'),

(3,10,'2025-06-03 00:00:00','pendiente'),

(4,9,'2025-06-04 00:00:00','entregado'),

(5,15,'2025-06-05 00:00:00','pendiente'),

(6,16,'2025-06-06 00:00:00','pendiente'),

(7,20,'2025-06-07 00:00:00','entregado'),

(8,16,'2025-06-08 00:00:00','pendiente'),

(9,12,'2025-06-09 00:00:00','pendiente'),

(10,6,'2025-06-10 00:00:00','entregado'),

(11,14,'2025-06-11 00:00:00','enviado'),

(12,18,'2025-06-12 00:00:00','pendiente'),

(13,17,'2025-06-13 00:00:00','pendiente'),

(14,16,'2025-06-14 00:00:00','enviado'),

(15,16,'2025-06-15 00:00:00','pendiente'),

(16,6,'2025-06-16 00:00:00','entregado'),

(17,5,'2025-06-17 00:00:00','entregado'),

(18,12,'2025-06-18 00:00:00','pendiente'),

(19,12,'2025-06-19 00:00:00','pendiente'),

(20,17,'2025-06-20 00:00:00','enviado');

INSERT INTO detalle\_pedido (id\_pedido, id\_producto, cantidad, total) VALUES

(9,1,4,3357496),

(16,4,4,3144756),

(14,3,5,2050985),

(19,3,2,820394),

(12,9,4,2972640),

(5,17,3,698232),

(4,13,4,2602800),

(16,2,2,889634),

(10,19,3,1292541),

(13,11,2,969842),

(16,13,1,650700),

(20,13,5,3253500),

(14,3,2,820394),

(17,16,4,1286148),

(3,11,4,1939684),

(20,8,4,3527392),

(4,11,5,2424605),

(16,2,5,2224085),

(12,10,4,2560900),

(20,12,4,1006128);

**3. PROCEDIMIENTO ALMACENADO PRINCIPAL:**

**· Desarrollar UN procedimiento que contenga TODAS las consultas SQL avanzadas**

**· Recibir parámetros para determinar tipo de consulta a ejecutar**

**Script:**

DELIMITER //

CREATE PROCEDURE ConsultasAvanzadas(

IN tipo\_operacion VARCHAR(50),

IN param1 VARCHAR(100),

IN param2 INT

)

BEGIN

CASE tipo\_operacion

-- CONSULTAS BÁSICAS

WHEN 'where' THEN

SELECT \* FROM productos WHERE precio > param2;

WHEN 'like' THEN

SELECT \* FROM productos WHERE nombre\_producto LIKE CONCAT('%', param1, '%');

WHEN 'order\_by' THEN

SELECT \* FROM productos ORDER BY precio DESC;

WHEN 'limit' THEN

SELECT \* FROM productos LIMIT param2;

WHEN 'in' THEN

SELECT \* FROM productos WHERE id\_producto IN (1, 2, 3);

WHEN 'distinct' THEN

SELECT DISTINCT nombre\_producto FROM productos;

-- AGREGACIONES Y GROUP BY

WHEN 'group\_by' THEN

SELECT id\_producto, SUM(cantidad) AS total\_vendido

FROM detalle\_pedido

GROUP BY id\_producto;

WHEN 'having' THEN

SELECT id\_producto, SUM(cantidad) AS total\_vendido

FROM detalle\_pedido

GROUP BY id\_producto

HAVING total\_vendido > param2;

-- JOINS

WHEN 'inner\_join' THEN

SELECT c.nombre, p.nombre\_producto

FROM clientes c

JOIN pedidos pe ON c.id\_cliente = pe.id\_cliente

JOIN detalle\_pedido dp ON pe.id\_pedido = dp.id\_pedido

JOIN productos p ON dp.id\_producto = p.id\_producto;

WHEN 'left\_join' THEN

SELECT c.nombre, p.nombre\_producto

FROM clientes c

LEFT JOIN pedidos pe ON c.id\_cliente = pe.id\_cliente

LEFT JOIN detalle\_pedido dp ON pe.id\_pedido = dp.id\_pedido

LEFT JOIN productos p ON dp.id\_producto = p.id\_producto;

-- SUBCONSULTAS

WHEN 'subquery' THEN

SELECT \* FROM productos

WHERE precio > (SELECT AVG(precio) FROM productos);

-- CTE + FUNCIONES DE VENTANA

WHEN 'cte\_window' THEN

WITH ranking AS (

SELECT id\_producto, nombre\_producto, precio,

ROW\_NUMBER() OVER (ORDER BY precio DESC) AS pos

FROM productos

)

SELECT \* FROM ranking WHERE pos <= param2;

-- FUNCIONES Y EXPRESIONES

WHEN 'case\_expr' THEN

SELECT nombre\_producto, precio,

CASE

WHEN precio < 80000 THEN 'Económico'

WHEN precio BETWEEN 80000 AND 130000 THEN 'Intermedio'

ELSE 'Premium'

END AS categoria

FROM productos;

-- MANIPULACIÓN DE STRINGS

WHEN 'string\_func' THEN

SELECT id\_producto, UPPER(nombre\_producto) AS nombre\_mayuscula

FROM productos;

-- MANIPULACIÓN DE FECHAS

WHEN 'date\_func' THEN

SELECT id\_producto, nombre\_producto, fecha\_creacion,

DATEDIFF(CURDATE(), fecha\_creacion) AS dias\_activo

FROM productos;

-- REGEX

WHEN 'regex' THEN

SELECT \* FROM productos

WHERE nombre\_producto REGEXP param1;

ELSE

SELECT 'Operación no válida.';

END CASE;

END //

DELIMITER ;

CALL ConsultasAvanzadas('where', '', 100000);

CALL ConsultasAvanzadas('like', 'HP', 0);

CALL ConsultasAvanzadas('order\_by', '', 0);

CALL ConsultasAvanzadas('limit', '', 3);

CALL ConsultasAvanzadas('in', '', 0);

CALL ConsultasAvanzadas('distinct', '', 0);

CALL ConsultasAvanzadas('group\_by', '', 0);

CALL ConsultasAvanzadas('having', '', 5);

CALL ConsultasAvanzadas('inner\_join', '', 0);

CALL ConsultasAvanzadas('left\_join', '', 0);

CALL ConsultasAvanzadas('subquery', '', 0);

CALL ConsultasAvanzadas('cte\_window', '', 3);

CALL ConsultasAvanzadas('case\_expr', '', 0);

CALL ConsultasAvanzadas('string\_func', '', 0);

CALL ConsultasAvanzadas('date\_func', '', 0);

CALL ConsultasAvanzadas('regex', 'Lenovo|HP', 0);

Conclusión

Nos vimos desafiados para generar el procedimiento almacenado de manera correcta, debido a las distintas posibilidades y cantidad de operaciones que pueden realizarse dentro del sistema.

A pesar de esto, continuamos implementando mejoras, hasta quedar felices con el resultado.

Aprendimos muchísimo con este proyecto, sin duda el desarrollo del procedimiento almacenado puso a prueba todo lo aprendido durante el semestre.