

ACTIVIDAD

1º Pastelería

DDL:

```
CREATE DATABASE Pasteleria  
USE Pasteleria;
```

```
CREATE TABLE Ingrediente (
```

id_ingredient	INT PRIMARY KEY AUTO_INCREMENT,
nombre_ingredient	VARCHAR (100) NOT NULL UNIQUE,
unidad_medida	VARCHAR (20) NOT NULL,
stock_actual	DECIMAL (10,2) NOT NULL DEFAULT 0,
costo_por_unidad	DECIMAL (10,2) NOT NULL

);

```
CREATE TABLE Producto (
```

id_producto	INT PRIMARY KEY AUTO_INCREMENT,
nombre_producto	VARCHAR (100) NOT NULL UNIQUE,
descripcion	TEXT,
precio_venta	DECIMAL (10,2) NOT NULL,
tiempo_preparacion_minutos	INT,
stock_disponible	INT NOT NULL DEFAULT 0

);

```
CREATE TABLE RECETA (
```

id_receta	INT PRIMARY KEY AUTO_INCREMENT,
id_producto	INT NOT NULL,
id_ingredient	INT NOT NULL,
cantidad_necesaria	DECIMAL (10,2) NOT NULL,
FOREIGN KEY (id_producto) REFERENCES Producto (id_producto),	
FOREIGN KEY (id_ingredient) REFERENCES Ingrediente (id_ingredient),	
UNIQUE (id_producto, id_ingredient)	

);

```
CREATE TABLE Cliente (
```

id_cliente	INT PRIMARY KEY AUTO_INCREMENT,
nombre_cliente	VARCHAR (50) NOT NULL,
apellido_cliente	VARCHAR (50) NOT NULL,
telefono_cliente	VARCHAR (20),
email_cliente	VARCHAR (100) UNIQUE,
direccion_cliente	VARCHAR (200)

);

CREATE TABLE Empleado (

id-empleado INT PRIMARY KEY AUTO-INCREMENT,
nombre-empleado VARCHAR (50) NOT Null,
apellido-empleado VARCHAR (50) NOT Null,
cargo VARCHAR (50),
fecha-contratacion DATE,
salario DECIMAL (10,2)
);

CREATE TABLE Pedido (

id-pedido INT PRIMARY KEY AUTO-INCREMENT,
fecha-pedido DATETIME NOT Null,
fecha-entrega-estimada DATETIME,
total-pedido DECIMAL (10,2) NOT Null,
estado-pedido VARCHAR (50) NOT Null,
id-Cliente INT,
id-empleado-recibe INT,
FOREIGN KEY (id-cliente) REFERENCES Cliente (id-cliente),
FOREIGN KEY (id-empleado-recibe) REFERENCES Empleado (id-empleado)
);

CREATE TABLE DetallePedido (

id-detalle-pedido INT PRIMARY KEY AUTO-INCREMENT,
id-pedido INT NOT Null,
id-producto INT NOT Null,
cantidad INT NOT Null,
precio-unitario-venta DECIMAL (10,2) NOT Null,
FOREIGN KEY (id-pedido) REFERENCES Pedido (id-pedido),
FOREIGN KEY (id-producto) REFERENCES Producto (id-producto),
UNIQUE (id-pedido, id-producto)
);

DML:

INSERT INTO Ingrediente (nombre-ingrediente, unidad-medida, stock-actual, costo-por-unidad) VALUES

('Harina de Trego', 'Kg', 50.00, 2.50),
('Azúcar blanca', 'Kg', 30.00, 1.80),
('Huevos', 'unidades', 200, 0.20),
('Mantequilla', 'Kg', 15.00, 8.00),
('Leche', 'litros', 25.00, 1.20),
('Chocolate Amargo', 'Kg', 10.00, 15.00),
('Fresas', 'Kg', 5.00, 6.00),
('Nueces', 'Kg', 8.00, 12.00);

INSERT INTO Producto (nombre_producto, descripcion, precio_venta, tiempo_preparacion_minutos, stock_disponible) **VALUES**

('Pastel de Chocolate', 'Delicioso pastel de chocolate con cobertura de ganache', 25.00, 60, 5),
('Tarta de Fresas', 'Tarta fresca con crema pastelera y fresas naturales', 22.50, 45, 3),
('Galletas de Mantecquilla', 'Galletas artesanales de mantecquilla (docena)', 10.00, 30, 20),
('Pan de Campo', 'Pan artesanal de masa madre', 80.00, 180, 10),
('Brownie con Nueces', 'Clasico Brownie fudgy con trozos de nueces', 3.50, 40, 15);

INSERT INTO Receta (id_producto, id_ingredient, cantidad_necesaria) **VALUES**

(1, 1, 0.50),
(1, 2, 0.40),
(1, 3, 4),
(1, 4, 0.25),
(1, 6, 0.30),
(2, 1, 0.30),
(2, 2, 0.20),
(2, 3, 2),
(2, 4, 0.15),
(2, 5, 0.50),
(2, 7, 0.50),
(3, 1, 0.20),
(3, 2, 0.15),
(3, 4, 0.20),
(5, 1, 0.15),
(5, 2, 0.30),
(5, 3, 3),
(5, 6, 0.25),
(5, 8, 0.10);

INSERT INTO Cliente (nombre_cliente, apellido_cliente, telefono_cliente, email_cliente, direccion_cliente) **VALUES**

('Ana', 'Garcia', '311234567', 'ana.garcia@example.com', 'Calle 10 # 5-15, Iquera'),
('Luis', 'Perez', '3229876543', 'luis.perez@example.com', 'Carrera 3 # 20-30, Neiva');

INSERT INTO Empleado (nombre_empleado, apellido_empleado, cargo, fecha_contratacion, salario) **VALUES**

('Carlos', 'Sanchez', 'Panadero Principal', '2023-01-10', 2500000.00),
('Marta', 'Diaz', 'Cajera', '2023-03-01', 1800000.00);

INSERT INTO Pedido (fecha_pedido, fecha_entrega_estimada, total_pedido, estado_pedido, id_cliente, id_empleado_recibe) **VALUES**

('2025-07-12 10:00:00', '2025-07-12 14:00:00', 25.00, 'Entregado', 1, 2),

('2025-07-12' 11:00:00', '2025-07-12' 15:00:00', 30.00,
'pediente', 2, 2);

INSERT INTO detallePedido (id_pedido, id_producto, cantidad,
precio_venta) VALUES
(1, 1, 1, 25.00),
(2, 3, 3, 10.00);

Consultas

1: SELECT

SELECT
nombre_producto,
precio_venta
FROM
Producto;

2: JOIN

SELECT
P.nombre_producto,
I.nombre_ingredientes,
R.cantidad_necesaria,
I.unidad_medida
FROM
Producto P
JOIN
Receta R ON P.id_producto = R.id_producto
JOIN
Ingrediente I ON R.id_ingredientes = I.id_ingredientes;

3: GROUP BY y COUNT

SELECT
E.nombre_empleado,
E.apellido_empleado,
COUNT(P.id_pedido) AS total_pedidos_tomados
FROM
Empleado E
LEFT JOIN
Pedido P ON E.id_empleado = P.id_empleado_recibe
GROUP BY
E.id_empleado, E.nombre_empleado, E.apellido_empleado
ORDER BY
total_pedidos_tomados DESC;

4. HAVING Y SUM

```
SELECT
    C.nombre_cliente,
    C.apellido_cliente,
    SUM (p.total_pedido) AS gasto_total
FROM
    cliente C
JOIN
    pedido p ON C.id_cliente = p.id_cliente
GROUP BY
    C.id_cliente, C.nombre_cliente, C.apellido_cliente
HAVING
    SUM (p.total_pedido) > 40;
```

5. LIMIT Y ORDER BY

```
SELECT
    nombre_producto,
    precio_venta
FROM
    producto
ORDER BY
    precio_venta DESC
LIMIT 3;
```

6. AVG

```
SELECT
    AVG (precio_venta) AS precio_promedio_productos
FROM
    producto;
```

7. DISTINCT

```
SELECT DISTINCT
    cargo
FROM
    Empleado;
```


8: subconsulta

```
SELECT
    nombre_cliente,
    apellido_cliente
FROM
    Cliente
WHERE
    id_cliente IN (
        SELECT p.id_cliente
        FROM Pedido P
        JOIN DetallePedido DP ON P.id_pedido = DP.id_pedido
        JOIN Producto Prod ON DP.id_producto = Prod.id_producto
        WHERE Prod.nombre_producto = 'pastel de chocolate'
    );
```

Sentencias

1: UPDATE

```
UPDATE Pedido
SET estado_pedido = 'Entregado'
    fecha-entrega-estimada = Now()
WHERE id_pedido = 2;

UPDATE Ingrediente
SET stock_actual = stock_actual + 10
WHERE nombre-ingrediente = 'Harina de trigo';
```

2: DELETE

```
DELETE FROM DetallePedido WHERE id_producto = 5;
DELETE FROM Receta WHERE id_producto = 5;
DELETE FROM Producto WHERE id_producto = 5;
```


Procedimiento to

DELIMITER //

CREATE PROCEDURE IniciarPreparacionPedido (

IN p-id-pedido INT,
OUT p-mensaje VARCHAR (255)

)

BEGIN

DECLARE v-estado-actual VARCHAR (30);
DECLARE v-id-producto INT;
DECLARE v-cantidad-producto INT;
DECLARE v-id-ingrediente INT;
DECLARE v-cantidad-necesaria-ingrediente DECIMAL (10,2);
DECLARE v-stock-ingrediente-actual DECIMAL (10,2);
DECLARE v-costo-ingredientes DECIMAL (10,2);
DECLARE done INT DEFAULT FALSE;

DECLARE cur-productos CURSOR FOR
SELECT dp.id-producto, dp.cantidad
FROM DetallePedido dp
WHERE dp.id-pedido = p-id-pedido;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

SELECT estado-pedido INTO v-estado-actual FROM pedido
WHERE id-pedido = p-id-pedido;

IF v-estado-actual IS NULL THEN
SET p-mensaje = 'Error: El ID de pedido especificado no
existe.';

ELSE IF v-estado-actual = 'En preparación' THEN
SET p-mensaje = 'El pedido ya está en preparación.';

ELSE IF v-estado-actual = 'Entregado' THEN
SET p-mensaje = 'El pedido ya ha sido entregado y no
puede ser modificado.';

ELSE

START TRANSACTION;
OPEN cur-productos;

read-loop: LOOP

FETCH cur-productos INTO v-id-productos, v-cantidad-producto

IF DONE THEN

LEAVE read-loop;

END IF;

BEGIN

DECLARE done-ingrediente INT DEFAULT FALSE;
DECLARE cur-ingredientes CURSOR FOR
SELECT R.id-ingrediente, R.cantidad-necesaria
FROM Receta R


```
WHERE R.id_producto = V.id_producto;  
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done_ingrediente  
= TRUE;
```

```
OPEN Cur_ingredientes;  
read_ingrediente_loop: loop  
FETCH Cur_ingredientes INTO V.id_ingrediente, V.cantidad-  
necesaria_ingrediente;  
IF done_ingrediente THEN  
LEAVE read_ingrediente_loop;  
END IF;
```

```
SET V.cantidad_necesaria_ingrediente = V.cantidad_necesaria-  
ingrediente * V.cantidad_producto;
```

```
SELECT stock_actual INTO V.stock_ingrediente_actual  
FROM Ingrediente  
WHERE id_ingrediente = V.id_ingrediente;
```

```
IF V.stock_ingrediente_actual < V.cantidad_necesaria-  
ingrediente THEN
```

```
ROLLBACK;
```

```
SET p-mensaje = CONCAT('Error: stock insuficiente para el  
ingrediente ID', V.id_ingrediente, '. Pedido no procesado.');
```

```
CLOSE Cur_ingredientes;
```

```
CLOSE Cur_productos;
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = p-mensaje;
```

```
END IF;
```

```
UPDATE Ingrediente
```

```
SET stock_actual = stock_actual - V.cantidad_necesaria_ingrediente  
WHERE id_ingrediente = V.id_ingrediente;
```

```
END loop read_ingrediente_loop;
```

```
CLOSE Cur_ingredientes;
```

```
END;
```

```
END loop read_loop;
```

```
CLOSE Cur_productos;
```

```
UPDATE Pedido
```

```
SET estado_pedido = 'En preparación'
```

```
WHERE id_pedido = p.id_pedido;
```

```
COMMIT
```

```
SET p-mensaje = CONCAT('El pedido', p.id_pedido, 'ha sido puesto  
en preparación y el stock de ingredientes actualizando.');
```

```
END IF;
```

```
END
```

```
SET @mensaje = '';
```

```
CALL IniciarPreparacionPedido (3, @mensaje);
```

```
SELECT @mensaje AS MensajeResultado;
```

```
SELECT id_pedido, estado_pedido FROM Pedido WHERE id_pedido = 3;  
SELECT nombre_ingrediente, stock_actual FROM Ingrediente WHERE  
nombre_ingrediente IN ('Harina', 'Azucar', 'Huevos', 'Mantecquilla', 'Leche');
```


Nombre de Tabla: Pastelería / Ingrediente

Fecha de Creación: 10/07/2025

Descripción: Po

Entidad que almacenará la información de los ingredientes

Clave

Campo

Tipo de Dato

Tamaño

Descripción

PK

id-ingrediente

INT

nombre-ingrediente

VARCHAR

100

unidad-medida

VARCHAR

20

stock-actual

DECIMAL

10,2

costo-por-unidad

DECIMAL

10,2

Identificador único de pastelería

Nombre del ingrediente

Unidad de medida del ingrediente

stock actual

Costo por unidad

Nombre de tabla: Producto

Fecha de creación: 10/07/2025

Descripción: En la tabla se almacena la información de los productos

La información de los productos

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-producto	INT		Identificador único de producto
	nombre-producto	VARCHAR	100	Nombre del producto
	descripcion	TEXT		Descripción del producto
	precio-venta	DECIMAL	10,2	Precio del producto
	tiempo-preparacion	INT		Tiempo de preparación
	stock-disponible	INT		Stock disponible

Nombre de Tabla: Receta

Fecha de Creación: 10/07/2025

Descripción: Entidad que almacena la información de la receta

Clave	Campo	Tipo de dato	Tamaño	Descripción
PK	id-receta	INT		Identificador único de receta
FK	id-producto	INT		
FK	id-ingrediente	INT		
	Cantidad-necesaria	DECIMAL	10,2	Cantidad necesaria de la receta

Nombre de Tabla: Cliente

Fecha de Creación: 10/07/2025

Descripción: Entidad que almacena la información personal del cliente

Clave	Campo	Tipo de dato	Tamaño	Descripción
PK	id_cliente	INT	-	Identificador único del cliente
	nombre_cliente	VARCHAR	50	Nombre del cliente
	apellido_cliente	VARCHAR	50	Apellido del cliente
	telefono_cliente	VARCHAR	20	Número de teléfono del cliente
	email_cliente	VARCHAR	100	email del cliente
	direccion_cliente	VARCHAR	200	Dirección en donde vive el cliente

Nombre de Tabla: Empleado

Fecha de creación: 10/07/2025

Descripción: Entidad que almacenará la información del empleado

Clave	Campo	Tipo de dato	Tamaño	Descripción
PK	id-empleado	INT		Identificador Único del empleado
	nombre-empleado	VARCHAR	50	Nombre del empleado
	apellido-empleado	VARCHAR	50	Apellido del empleado
	Cargo	VARCHAR	50	Cargo que tiene el empleado
	fecha-contratación	DATE		Fecha de contratación
	salario	DECIMAL	10,2	Salario del empleado

Nombre de Tabla: Pedido

Fecha de creación: 10/04/2025

Descripción: Entidad que almacenará la información del pedido

Clave	Campo	tipo de dato	Tamaño	Descripción
PK	id_pedido	INT		Identificador único del pedido
	fecha_pedido	DATE TIME		Fecha del pedido
	fecha_entrega	DATE TIME		Fecha de entrega de entrega
	total_pedido	DECIMAL	10,2	Total de pedido
	estado_pedido	VARCHAR	50	Estado del pedido
PK	id_cliente	INT		
PK	id_empleado	INT		

Nombre de Tabla: DetallePedido

Fecha de creación: 10/07/2025

Descripción: Cantidad que almacenará la información de Detalle Pedido

Clave	Campo	Tipo de dato	Tamaño	Descripción
PK	id-detalle_pedido	INT		Identificador único de detalle pedido
FK	id_pedido	INT		
FK	id_producto	INT		
	Cantidad	INT		Cantidad de detalle pedido
	Precio_unitario	DECIMAL	10, 2	Precio unitario de detalle pedido

Ingrediente
id-ingrediente (PK)
nombre-ingrediente
unidad-medida
stock-actual
costo-por-unidad

Producto
id-Producto (PK)
nombre-Producto
descripcion
precio-venta
tempo-preparacion-menos
stock-disponible

Cliente
id-cliente
nombre-cliente
apellido-cliente
telefono-cliente
email-cliente
direccion-cliente

Detalle Pedido
id-detalle-pedido (PK)
id-pedido (FK)
id-Producto (FK)
precio-unitario-venta
cantidad

ReCeta
id-receta (PK)
id-Producto (FK)
id-ingrediente (FK)
cantidad-receta

Empleado
id-empleado (PK)
nombre-empleado
apellido-empleado
cargo
fecha-contratacion
salario

Pedido
id-pedido (PK)
fecha-pedido
fecha-entrega-estimada
total-pedido
estado-pedido
id-cliente (FK)
id-empleado (FK)

N:1

1:1

1:1

N:1

N:M

N:1