

Este taller está diseñado para profundizar en el manejo y optimización de bases de datos MySQL.

A través de ejercicios prácticos, se explorarán temas avanzados para reforzar el conocimiento en

normalización, joins, consultas complejas, subconsultas, procedimientos almacenados, funciones

definidas por el usuario y triggers.

Requisitos previos:

Conocimiento básico de SQL y MySQL

MySQL instalado y configurado en tu máquina

Objetivos:

Al finalizar este taller, el participante será capaz de:

1. Diseñar bases de datos optimizadas mediante técnicas de normalización.
2. Realizar consultas avanzadas en múltiples tablas.
3. Utilizar subconsultas para consultas complejas.
4. Crear y ejecutar procedimientos almacenados y funciones definidas por el usuario.
5. Implementar triggers para automatizar operaciones en la base de datos.

No se encontró "Pasted image 20250621194351.png".

```
-- Creación de la base de datos
CREATE DATABASE vtaszfs;
USE vtaszfs;
-- Tabla Clientes
CREATE TABLE Clientes (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    email VARCHAR(100) UNIQUE
);
-- Tabla UbicacionCliente
CREATE TABLE UbicacionCliente (
    id INT PRIMARY KEY AUTO_INCREMENT,
    cliente_id INT,
    direccion VARCHAR(255),
    ciudad VARCHAR(100),
    estado VARCHAR(50),
    codigo_postal VARCHAR(10),
    pais VARCHAR(50),
```

```

        FOREIGN KEY (cliente_id) REFERENCES Clientes(id)
    );

-- Tabla Empleados
CREATE TABLE Empleados (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    puesto VARCHAR(50),
    salario DECIMAL(10, 2),
    fecha_contratacion DATE
);

-- Tabla Proveedores
CREATE TABLE Proveedores (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    contacto VARCHAR(100),
    telefono VARCHAR(20),
    direccion VARCHAR(255)
);

-- Tabla TiposProductos
CREATE TABLE TiposProductos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    tipo_nombre VARCHAR(100),
    descripcion TEXT
);

-- Tabla Productos
CREATE TABLE Productos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    precio DECIMAL(10, 2),
    proveedor_id INT,
    tipo_id INT,
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(id),
    FOREIGN KEY (tipo_id) REFERENCES TiposProductos(id)
);

-- Tabla Pedidos
CREATE TABLE Pedidos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    cliente_id INT,
    fecha DATE,
    total DECIMAL(10, 2),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(id)
);

-- Tabla DetallesPedido
CREATE TABLE DetallesPedido (
    id INT PRIMARY KEY AUTO_INCREMENT,

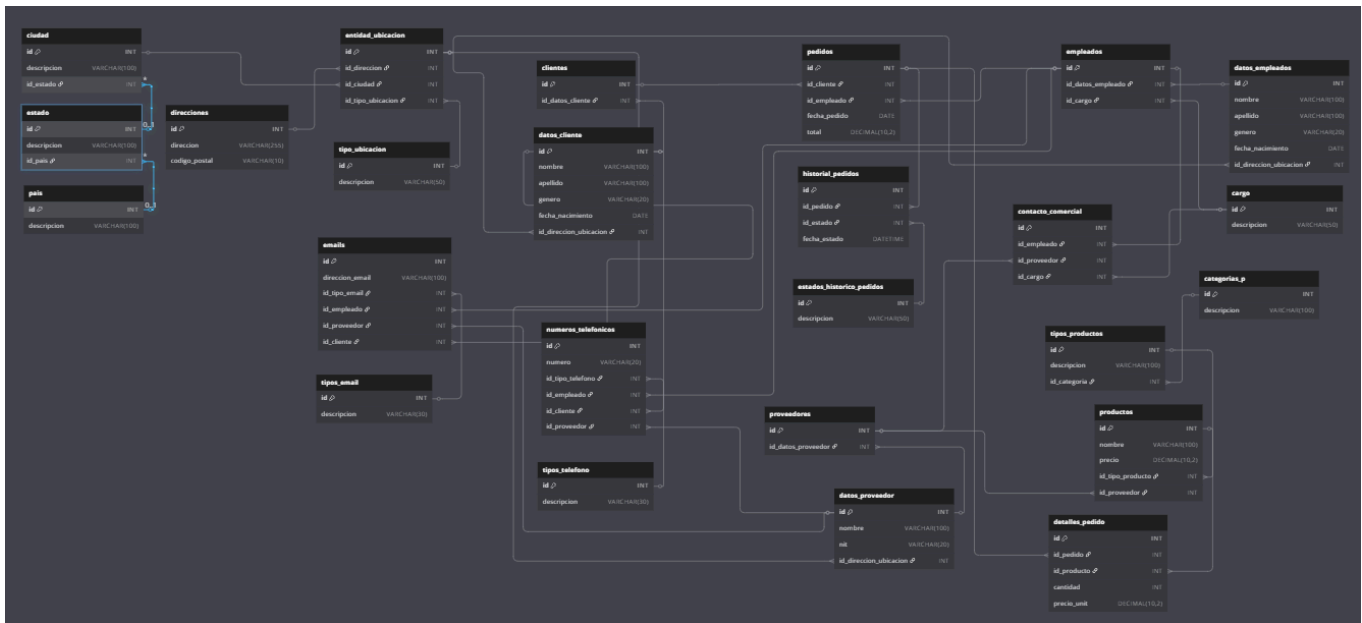
```

```

pedido_id INT,
producto_id INT,
cantidad INT,
precio DECIMAL(10, 2),
FOREIGN KEY (pedido_id) REFERENCES Pedidos(id),
FOREIGN KEY (producto_id) REFERENCES Productos(id)
);

```

## Normalización



```

CREATE DATABASE IF NOT EXISTS vtaszfs;
USE vtaszfs;

```

```

CREATE TABLE pais (
    id INT PRIMARY KEY AUTO_INCREMENT,
    descripcion VARCHAR(100)
);

```

```

CREATE TABLE estado (
    id INT PRIMARY KEY AUTO_INCREMENT,
    descripcion VARCHAR(100),
    id_pais INT,
    FOREIGN KEY (id_pais) REFERENCES pais(id)
);

```

```

CREATE TABLE ciudad (
    id INT PRIMARY KEY AUTO_INCREMENT,
    descripcion VARCHAR(100),

```

```
    id_estado INT,  
    FOREIGN KEY (id_estado) REFERENCES estado(id)  
);  
  
CREATE TABLE direcciones (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    direccion VARCHAR(255),  
    codigo_postal VARCHAR(10)  
);  
  
CREATE TABLE tipo_ubicacion (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(50)  
);  
  
CREATE TABLE entidad_ubicacion (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    id_direccion INT,  
    id_ciudad INT,  
    id_tipo_ubicacion INT,  
    FOREIGN KEY (id_direccion) REFERENCES direcciones(id),  
    FOREIGN KEY (id_ciudad) REFERENCES ciudad(id),  
    FOREIGN KEY (id_tipo_ubicacion) REFERENCES tipo_ubicacion(id)  
);  
  
CREATE TABLE cargo (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(50)  
);  
  
CREATE TABLE categorias_p (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(100)  
);  
  
CREATE TABLE tipos_email (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(30)  
);  
  
CREATE TABLE tipos_telefono (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(30)  
);  
  
CREATE TABLE estados_historico_pedidos (  

```

```

    id INT PRIMARY KEY AUTO_INCREMENT,
    descripcion VARCHAR(50)
);

CREATE TABLE datos_cliente (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    apellido VARCHAR(100),
    genero VARCHAR(20),
    fecha_nacimiento DATE,
    id_direccion_ubicacion INT,
    FOREIGN KEY (id_direccion_ubicacion) REFERENCES entidad_ubicacion(id)
);

CREATE TABLE datos_empleados (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    apellido VARCHAR(100),
    genero VARCHAR(20),
    fecha_nacimiento DATE,
    id_direccion_ubicacion INT,
    FOREIGN KEY (id_direccion_ubicacion) REFERENCES entidad_ubicacion(id)
);

CREATE TABLE datos_proveedor (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    nit VARCHAR(20),
    id_direccion_ubicacion INT,
    FOREIGN KEY (id_direccion_ubicacion) REFERENCES entidad_ubicacion(id)
);

CREATE TABLE clientes (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_datos_cliente INT,
    FOREIGN KEY (id_datos_cliente) REFERENCES datos_cliente(id)
);

CREATE TABLE empleados (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_datos_empleado INT,
    id_cargo INT,
    FOREIGN KEY (id_datos_empleado) REFERENCES datos_empleados(id),
    FOREIGN KEY (id_cargo) REFERENCES cargo(id)
);

```

```
CREATE TABLE proveedores (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    id_datos_proveedor INT,  
    FOREIGN KEY (id_datos_proveedor) REFERENCES datos_proveedor(id)  
);
```

```
CREATE TABLE emails (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    direccion_email VARCHAR(100),  
    id_tipo_email INT,  
    id_empleado INT,  
    id_proveedor INT,  
    id_cliente INT,  
    FOREIGN KEY (id_tipo_email) REFERENCES tipos_email(id),  
    FOREIGN KEY (id_empleado) REFERENCES empleados(id),  
    FOREIGN KEY (id_proveedor) REFERENCES datos_proveedor(id),  
    FOREIGN KEY (id_cliente) REFERENCES datos_cliente(id)  
);
```

```
CREATE TABLE numeros_telefonicos (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    numero VARCHAR(20),  
    id_tipo_telefono INT,  
    id_empleado INT,  
    id_cliente INT,  
    id_proveedor INT,  
    FOREIGN KEY (id_tipo_telefono) REFERENCES tipos_telefono(id),  
    FOREIGN KEY (id_empleado) REFERENCES empleados(id),  
    FOREIGN KEY (id_cliente) REFERENCES datos_cliente(id),  
    FOREIGN KEY (id_proveedor) REFERENCES datos_proveedor(id)  
);
```

```
CREATE TABLE tipos_productos (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    descripcion VARCHAR(100),  
    id_categoria INT,  
    FOREIGN KEY (id_categoria) REFERENCES categorias_p(id)  
);
```

```
CREATE TABLE productos (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    precio DECIMAL(10,2),  
    id_tipo_producto INT,  
    id_proveedor INT,  
    FOREIGN KEY (id_tipo_producto) REFERENCES tipos_productos(id),
```

```

    FOREIGN KEY (id_proveedor) REFERENCES proveedores(id)
);

CREATE TABLE pedidos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_cliente INT,
    id_empleado INT,
    fecha_pedido DATE,
    total DECIMAL(10,2),
    FOREIGN KEY (id_cliente) REFERENCES clientes(id),
    FOREIGN KEY (id_empleado) REFERENCES empleados(id)
);

CREATE TABLE detalles_pedido (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_pedido INT,
    id_producto INT,
    cantidad INT,
    precio_unit DECIMAL(10,2),
    FOREIGN KEY (id_pedido) REFERENCES pedidos(id),
    FOREIGN KEY (id_producto) REFERENCES productos(id)
);

CREATE TABLE historial_pedidos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_pedido INT,
    id_estado INT,
    fecha_estado DATETIME,
    FOREIGN KEY (id_pedido) REFERENCES pedidos(id),
    FOREIGN KEY (id_estado) REFERENCES estados_historico_pedidos(id)
);

CREATE TABLE contacto_comercial (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_empleado INT,
    id_proveedor INT,
    id_cargo INT,
    FOREIGN KEY (id_empleado) REFERENCES empleados(id),
    FOREIGN KEY (id_proveedor) REFERENCES proveedores(id),
    FOREIGN KEY (id_cargo) REFERENCES cargo(id)
);

```

## INSERCIÓN DE DATOS

## TABLA pais

```
INSERT INTO Pais (descripcion) VALUES ('Colombia');
INSERT INTO Pais (descripcion) VALUES ('Estados_unidos');
INSERT INTO Pais (descripcion) VALUES ('Panama');
```

## TABLA estado

```
INSERT INTO Estado (descripcion, id_pais) VALUES
('Amazonas', 1),('Antioquia', 1),('Arauca', 1),('Atlántico', 1),('Bogotá
D.C.', 1),('Bolívar', 1),('Boyacá', 1),('Caldas', 1),('Caquetá', 1),
('Casanare', 1),('Cauca', 1),('Cesar', 1),('Chocó', 1),('Córdoba', 1),
('Cundinamarca', 1),('Guainía', 1),('Guaviare', 1),('Huila', 1),('La Guajira',
1),('Magdalena', 1),('Meta', 1),('Nariño', 1),('Norte de Santander', 1),
('Putumayo', 1),('Quindío', 1),('Risaralda', 1),('San Andrés y Providencia',
1),('Santander', 1),('Sucre', 1),('Tolima', 1),('Valle del Cauca', 1),
('Vaupés', 1),('Vichada', 1),('Alabama', 2),('Alaska', 2),('Arizona', 2),
('Arkansas', 2),('California', 2),('Colorado', 2),('Connecticut', 2),
('Delaware', 2),('Florida', 2),('Georgia', 2),('Hawaii', 2),('Idaho', 2),
('Illinois', 2),('Indiana', 2),('Iowa', 2),('Kansas', 2),('Kentucky', 2),
('Louisiana', 2),('Maine', 2),('Maryland', 2),('Massachusetts', 2),
('Michigan', 2),('Minnesota', 2),('Mississippi', 2),('Missouri', 2),
('Montana', 2),('Nebraska', 2),('Nevada', 2),('New Hampshire', 2),('New
Jersey', 2),('New Mexico', 2),('New York', 2),('North Carolina', 2),('North
Dakota', 2),('Ohio', 2),('Oklahoma', 2),('Oregon', 2),('Pennsylvania', 2),
('Rhode Island', 2),('South Carolina', 2),('South Dakota', 2),('Tennessee',
2),('Texas', 2),('Utah', 2),('Vermont', 2),('Virginia', 2),('Washington', 2),
('West Virginia', 2),('Wisconsin', 2),('Wyoming', 2);
```

## TABLA ciudad

```
INSERT INTO Ciudad (descripcion, id_estado) VALUES
('Leticia', 1), ('Medellín', 2), ('Arauca', 3),('Barranquilla', 4),('Bogotá',
5),
('Cartagena', 6),('Tunja', 7),('Manizales', 8),('Florencia', 9),('Yopal', 10),
('Popayán', 11),('Valledupar', 12),('Quibdó', 13),('Montería', 14),
('Girardot', 15),('Inírida', 16),('San José del Guaviare', 17),('Neiva', 18),
('Riohacha', 19),('Santa Marta', 20),('Villavicencio', 21),('Pasto', 22),
('Cúcuta', 23),('Mocoa', 24),('Armenia', 25),('Pereira', 26),('San Andrés',
27),('Bucaramanga', 28),('Sincelejo', 29),('Ibagué', 30),('Cali', 31),('Mitú',
32),('Puerto Carreño', 33),
('Birmingham', 34), ('Anchorage', 35), ('Phoenix', 36), ('Little Rock', 37),
('Los Angeles', 38), ('Denver', 39), ('Bridgeport', 40), ('Wilmington', 41),
('Jacksonville', 42), ('Atlanta', 43), ('Honolulu', 44), ('Boise', 45),
```



```
( 'Chicago', 46), ( 'Indianapolis', 47), ( 'Des Moines', 48), ( 'Wichita', 49),
( 'Louisville', 50), ( 'New Orleans', 51), ( 'Portland', 52), ( 'Baltimore', 53),
( 'Boston', 54), ( 'Detroit', 55), ( 'Minneapolis', 56), ( 'Jackson', 57),
( 'Kansas City', 58), ( 'Billings', 59), ( 'Omaha', 60), ( 'Las Vegas', 61),
( 'Manchester', 62), ( 'Newark', 63), ( 'Albuquerque', 64), ( 'New York City',
65), ( 'Charlotte', 66), ( 'Fargo', 67), ( 'Columbus', 68), ( 'Oklahoma City',
69), ( 'Portland', 70), ( 'Philadelphia', 71), ( 'Providence', 72),
( 'Charleston', 73), ( 'Sioux Falls', 74), ( 'Nashville', 75), ( 'Houston', 76),
( 'Salt Lake City', 77), ( 'Burlington', 78), ( 'Virginia Beach', 79),
( 'Seattle', 80), ( 'Charleston', 81), ( 'Milwaukee', 82), ( 'Cheyenne', 83);
```

```
INSERT INTO Direcciones (direccion, codigo_postal) VALUES
("Calle 10 #5-21", "110111"),("Carrera 7 #32-15", "110221"),("Avenida Caracas
#17-23", "110311"),("Transversal 23 #45-67", "110321"),("Diagonal 40A #23-45",
"110411"),("Calle 72 #10-20", "110121"),("Carrera 30 #20-50", "110231"),
("Avenida Suba #100-45", "111121"),("Calle 26 #68D-35", "110931"),("Calle 63
#24-60", "110821"),("Carrera 50 #60-20", "110711"),("Avenida Boyacá #56-78",
"111021"),("Calle 80 #70-20", "111011"),("Carrera 15 #85-30", "110911"),
("Avenida 68 #22-10", "110511");
```

```
INSERT INTO Entidad_Ubicacion (id_direccion, id_ciudad) VALUES
(1, 5),(2, 17),(3, 3),(4, 12),(5, 8),(6, 21),(7, 2),(8, 14),(9, 29),(10, 9),
(11, 25),(12, 6),(13, 18),(14, 30),(15, 11);
```

```
INSERT INTO Categorías_P (descripcion) VALUES
('Tecnología'), ('Electrodomésticos'), ('Hogar_Decoración'),
('Ropa_Accesorios'), ('Salud_Belleza'), ('Deportes_Aire Libre'),
('Juguetes_Juegos'), ('Automotriz'), ('Libros'), ('Música'), ('Cine_Series'),
('Computación'), ('Celulares_Tablets'), ('Videojuegos'), ('Herramientas'),
('Jardinería'), ('Mascotas'), ('Papelería_Oficina'), ('Alimentos_Bebidas'),
('Servicios');
```

```
INSERT INTO Tipos_Productos (descripcion, id_categoria) VALUES
('Computadores', 1), ('Smartphones', 1), ('Tablets', 1), ('Refrigeradoras',
2), ('Lavadoras', 2), ('Microondas', 2), ('Sofás', 3), ('Lámparas', 3),
('Cuadros_Decorativos', 3), ('Camisetas', 4), ('Pantalones', 4), ('Zapatos',
4), ('Vitaminas', 5), ('Maquillaje', 5), ('Cremas_Corporales', 5),
('Bicicletas', 6), ('Balones', 6), ('Tiendas_de_campaña', 6), ('Muñecas', 7),
('Juegos_de_mesa', 7), ('Puzzles', 7), ('Llantas', 8), ('Aceite_de_motor', 8),
('GPS_para_autos', 8), ('Novelas', 9), ('Libros_técnicos', 9),
('Cuentos_infantiles', 9), ('CDs', 10), ('Vinilos', 10),
('Accesorios_de_instrumentos', 10), ('DVDs', 11), ('Blu-ray', 11),
```

```
('Posters_de_películas', 11), ('Teclados', 12), ('Mouses', 12), ('Monitores', 12), ('Fundas', 13), ('Cargadores', 13), ('Protectores_de_pantalla', 13), ('Consolas', 14), ('Controles', 14), ('Juegos_digitales', 14), ('Taladros', 15), ('Destornilladores', 15), ('Lijadoras', 15), ('Macetas', 16), ('Tierra_abonada', 16), ('Semillas', 16), ('Comida_para_perros', 17), ('Juguetes_para_gatos', 17), ('Arena_sanitaria', 17), ('Cuadernos', 18), ('Bolígrafos', 18), ('Carpetas', 18), ('Snacks', 19), ('Bebidas_energéticas', 19), ('Dulces', 19), ('Asesorías', 20), ('Instalaciones', 20), ('Reparaciones', 20)
```

```
INSERT INTO Tipos_Telefono (descripcion) VALUES ('Móvil'), ('Fijo');
INSERT INTO Tipos_Email (descripcion) VALUES ('Corporativo'), ('Personal');
INSERT INTO Estados_Historico_Pedidos (descripcion) VALUES ('Pendiente'), ('Enviado'), ('Entregado');
INSERT INTO id_fiscal (descripcion) VALUES ('natural'), ('juridico');
INSERT INTO Cargo (descripcion) VALUES ('Vendedor'), ('Administrador'), ('Gerente'), ('Asistente'), ('Supervisor');
INSERT INTO Tipo_Ubicacion (descripcion) VALUES ('Cliente'), ('Empleado'), ('Proveedor');
```

```
INSERT INTO Datos_Proveedor (nombre, nit, id_direccion_ubicacion) VALUES ('Distribuidora Andina S.A.S', '900111001', 16), ('Proveedores del Norte LTDA', '900111002', 17), ('Suministros Globales S.A.', '900111003', 18), ('Importadora del Centro', '900111004', 19), ('TechMarket Colombia', '900111005', 20)
```

```
INSERT INTO Proveedores (id_datos_proveedor) VALUES (11), (12), (13), (14), (15);
```

```
INSERT INTO Datos_Empleados (nombre, apellido, genero, fecha_nacimiento, id_direccion_ubicacion)
VALUES ('Carlos', 'Ramírez', 'Masculino', '1990-05-12', 21), ('María', 'López', 'Femenino', '1985-08-23', 22), ('Andrés', 'González', 'Masculino', '1992-11-04', 23), ('Luisa', 'Martínez', 'Femenino', '1988-01-16', 24), ('Juan', 'Torres', 'Masculino', '1995-03-30', 25);
```

```
INSERT INTO Empleados (id_datos_empleado, id_cargo)VALUES  
(6, 1),(7, 2),(8, 3),(9, 4),(10, 5);
```

```
INSERT INTO Datos_Cliente (nombre, apellido, genero, fecha_nacimiento,  
id_direccion_ubicacion)  
VALUES  
( 'María', 'Gómez', 'Femenino', '1992-08-25', 26),  
( 'Jorge', 'Ríos', 'Masculino', '1989-04-15', 27),  
( 'Ana', 'Castro', 'Femenino', '1995-12-03', 28),  
( 'Luis', 'Fernández', 'Masculino', '1991-07-10', 29),  
( 'Paola', 'Mejía', 'Femenino', '1993-11-19', 30);
```

```
INSERT INTO Clientes (id_datos_cliente) VALUES  
(1),(2),(3),(4),(5);
```

```
INSERT INTO Productos (nombre, precio, id_tipo_producto, id_proveedor) VALUES  
( 'Laptop Lenovo Ideapad 3', 3200000, 1, 17),  
( 'Mouse Logitech M185', 65000, 35, 17),  
( 'Smartphone Samsung A34', 1450000, 2, 18),  
( 'Cargador Samsung 25W', 120000, 38, 18),  
( 'Refrigeradora LG 380L', 2750000, 4, 19),  
( 'Microondas Haceb 20L', 380000, 6, 19),  
( 'Sofá en L Gris Oscuro', 2100000, 7, 20),  
( 'Lámpara LED Escritorio', 89000, 8, 20),  
( 'Camiseta Nike Dri-FIT', 95000, 10, 21),  
( 'Zapatos Adidas Runfalcon', 220000, 12, 21);
```

```
INSERT INTO Pedidos (id_cliente, id_empleado, fecha_pedido, total) VALUES  
(6, 6, '2025-06-23', 3200000),(6, 6, '2025-06-22', 65000),(6, 6, '2025-06-21',  
1450000),(7, 7, '2025-06-23', 120000),(7, 7, '2025-06-22', 2750000),(7, 7,  
'2025-06-21', 380000),(8, 8, '2025-06-23', 2100000),(8, 8, '2025-06-22',  
89000),(8, 8, '2025-06-21', 95000),(9, 9, '2025-06-23', 220000),(9, 9, '2025-  
06-22', 3200000),(9, 9, '2025-06-21', 1450000),(10, 10, '2025-06-23', 120000),  
(10, 10, '2025-06-22', 2750000),(10, 10, '2025-06-21', 380000);
```

```
INSERT INTO Historial_Pedidos (id_pedido, id_estado, fecha_estado) VALUES  
(1, 1, NOW()), (2, 1, NOW()), (3, 1, NOW()),  
(4, 1, NOW()), (5, 1, NOW()), (6, 1, NOW()),  
(7, 1, NOW()), (8, 1, NOW()), (9, 1, NOW()),
```

```
(10, 1, NOW()), (11, 1, NOW()), (12, 1, NOW()),  
(13, 1, NOW()), (14, 1, NOW()), (15, 1, NOW());
```

```
INSERT INTO Emails (direccion_email, id_tipo_email, id_cliente)VALUES  
( 'maria.gomez@gmail.com', 1, 1),  
( 'jorge.rios@hotmail.com', 1, 2),  
( 'ana.castro@yahoo.com', 1, 3),  
( 'luis.fernandez@outlook.com', 1, 4),  
( 'paola.mejia@gmail.com', 1, 5),  
( 'carlos.ramirez@empresa.com', 2, 6),  
( 'maria.lopez@empresa.com', 2, 7),  
( 'andres.gonzalez@empresa.com', 2, 8),  
( 'luisa.martinez@empresa.com', 2, 9),  
( 'juan.torres@empresa.com', 2, 10),  
( 'contacto@andina.com', 2, 11),  
( 'info@proveedoresnorte.com', 2, 12),  
( 'ventas@suministrosglobales.com', 2, 13),  
( 'importadora@centro.com', 2, 14),  
( 'contacto@techmarket.com', 2, 15);
```

```
INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad, precio_unit)  
VALUES  
(1, 1, 2, 15990.00),(2, 3, 1, 20900.00),(3, 5, 3, 9900.00),(4, 6, 2,  
18900.00),(5, 2, 1, 25900.00),(6, 4, 4, 12990.00),(7, 7, 1, 14900.00),(8, 8,  
2, 18990.00),(9, 9, 2, 17900.00),(10, 10, 3, 9900.00);
```

```
INSERT INTO contacto_comercial (id_empleado, id_proveedor, id_cargo)  
VALUES  
(7, 17, 2),(7, 18, 2),(7, 19, 2),(9, 20, 4),(9, 21, 4);
```

```
INSERT INTO Numeros_Telefonicos (numero, id_tipo_telefono,  
id_entidad_ubicacion)  
VALUES  
( '3001110001', 1, 16),( '3001110002', 1, 17),( '3001110003', 1, 18),  
( '3001110004', 1, 19),( '3001110005', 1, 20),( '3012220001', 1, 21),  
( '3012220002', 1, 22),( '3012220003', 1, 23),( '3023330001', 1, 24),  
( '3023330002', 1, 25),( '3023330003', 1, 26),( '3023330004', 1, 27),  
( '3023330005', 1, 28),( '3023330006', 1, 29),( '3023330007', 1, 30);
```

```

INSERT INTO Emails (email, id_tipo_email, id_entidad_ubicacion)
VALUES
('proveedor1@empresa.com', 1, 16),('proveedor2@empresa.com', 1, 17),
('proveedor3@empresa.com', 1, 18),('proveedor4@empresa.com', 1, 19),
('proveedor5@empresa.com', 1, 20),('empleado1@empresa.com', 1, 21),
('empleado2@empresa.com', 1, 22),('cliente1@gmail.com', 2, 24),
('cliente2@gmail.com', 2, 25),('cliente3@gmail.com', 2, 26),
('cliente4@gmail.com', 2, 27),('cliente5@gmail.com', 2, 28),
('cliente6@gmail.com', 2, 29),('cliente7@gmail.com', 2, 30);

```

## Joins

1. Obtener la lista de todos los pedidos con los nombres de clientes usando INNER JOIN .

```

mysql> SELECT
->     p.id AS id_pedido,
->     dc.nombre AS nombre_cliente,
->     dc.apellido AS apellido_cliente,
->     p.fecha_pedido,
->     p.total
-> FROM pedidos p
-> INNER JOIN clientes c ON p.id_cliente = c.id
-> INNER JOIN datos_cliente dc ON c.id_datos_cliente = dc.id;

```

id_pedido	nombre_cliente	apellido_cliente	fecha_pedido	total
1	María	Gómez	2025-06-23	3200000.00
2	María	Gómez	2025-06-22	65000.00
3	María	Gómez	2025-06-21	1450000.00
4	Jorge	Ríos	2025-06-23	120000.00
5	Jorge	Ríos	2025-06-22	2750000.00
6	Jorge	Ríos	2025-06-21	380000.00
7	Ana	Castro	2025-06-23	2100000.00
8	Ana	Castro	2025-06-22	89000.00
9	Ana	Castro	2025-06-21	95000.00
10	Luis	Fernández	2025-06-23	220000.00
11	Luis	Fernández	2025-06-22	3200000.00
12	Luis	Fernández	2025-06-21	1450000.00
13	Paola	Mejía	2025-06-23	120000.00
14	Paola	Mejía	2025-06-22	2750000.00
15	Paola	Mejía	2025-06-21	380000.00

15 rows in set (0.00 sec)

## 2. Listar los productos y proveedores que los suministran con INNER JOIN .

```
mysql> SELECT
->     p.id AS id_producto,
->     p.nombre AS nombre_producto,
->     dp.nombre AS nombre_proveedor
-> FROM productos p
-> INNER JOIN proveedores pr ON p.id_proveedor = pr.id
-> INNER JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id;
```

id_producto	nombre_producto	nombre_proveedor
1	Laptop Lenovo Ideapad 3	Distribuidora Andina S.A.S
2	Mouse Logitech M185	Distribuidora Andina S.A.S
3	Smartphone Samsung A34	Proveedores del Norte LTDA
4	Cargador Samsung 25W	Proveedores del Norte LTDA
5	Refrigeradora LG 380L	Suministros Globales S.A.
6	Microondas Haceb 20L	Suministros Globales S.A.
7	Sofá en L Gris Oscuro	Importadora del Centro
8	Lámpara LED Escritorio	Importadora del Centro
9	Camiseta Nike Dri-FIT	TechMarket Colombia
10	Zapatos Adidas Runfalcon	TechMarket Colombia

10 rows in set (0.00 sec)

### 3. Mostrar los pedidos y las ubicaciones de los clientes con LEFT JOIN .

```
mysql> SELECT
-> p.id AS id_pedido,
-> dc.nombre AS nombre_cliente,
-> dc.apellido AS apellido_cliente,
-> ci.descripcion AS ciudad,
-> es.descripcion AS estado,
-> pa.descripcion AS pais,
-> p.fecha_pedido,
-> p.total
-> FROM pedidos p
-> LEFT JOIN clientes c ON p.id_cliente = c.id
-> LEFT JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> LEFT JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
-> LEFT JOIN ciudad ci ON eu.id_ciudad = ci.id
-> LEFT JOIN estado es ON ci.id_estado = es.id
-> LEFT JOIN pais pa ON es.id_pais = pa.id;
```

id_pedido	nombre_cliente	apellido_cliente	ciudad	estado	pais	fecha_pedido	total
1	María	Gómez	Armenia	Quindío	Colombia	2025-06-23	3200000.00
2	María	Gómez	Armenia	Quindío	Colombia	2025-06-22	65000.00
3	María	Gómez	Armenia	Quindío	Colombia	2025-06-21	1450000.00
4	Jorge	Ríos	Cartagena	Bolívar	Colombia	2025-06-23	120000.00
5	Jorge	Ríos	Cartagena	Bolívar	Colombia	2025-06-22	2750000.00
6	Jorge	Ríos	Cartagena	Bolívar	Colombia	2025-06-21	380000.00
7	Ana	Castro	Neiva	Huila	Colombia	2025-06-23	2100000.00
8	Ana	Castro	Neiva	Huila	Colombia	2025-06-22	89000.00
9	Ana	Castro	Neiva	Huila	Colombia	2025-06-21	95000.00
10	Luis	Fernández	Ibagué	Tolima	Colombia	2025-06-23	220000.00
11	Luis	Fernández	Ibagué	Tolima	Colombia	2025-06-22	3200000.00
12	Luis	Fernández	Ibagué	Tolima	Colombia	2025-06-21	1450000.00
13	Paola	Mejía	Popayán	Cauca	Colombia	2025-06-23	120000.00
14	Paola	Mejía	Popayán	Cauca	Colombia	2025-06-22	2750000.00
15	Paola	Mejía	Popayán	Cauca	Colombia	2025-06-21	380000.00

### 4. Consultar los empleados que han registrado pedidos, incluyendo empleados sin pedidos ( LEFT JOIN ).

```
mysql> SELECT
-> e.id AS id_empleado,
-> de.nombre AS nombre_empleado,
-> de.apellido AS apellido_empleado,
-> p.id AS id_pedido,
-> p.fecha_pedido,
-> p.total
-> FROM empleados e
-> LEFT JOIN datos_empleados de ON e.id_datos_empleado = de.id
-> LEFT JOIN pedidos p ON e.id = p.id_empleado;
```

id_empleado	nombre_empleado	apellido_empleado	id_pedido	fecha_pedido	total
6	Carlos	Ramírez	1	2025-06-23	3200000.00
6	Carlos	Ramírez	2	2025-06-22	65000.00
6	Carlos	Ramírez	3	2025-06-21	1450000.00
7	María	López	4	2025-06-23	120000.00
7	María	López	5	2025-06-22	2750000.00
7	María	López	6	2025-06-21	380000.00
8	Andrés	González	7	2025-06-23	2100000.00
8	Andrés	González	8	2025-06-22	89000.00
8	Andrés	González	9	2025-06-21	95000.00
9	Luisa	Martínez	10	2025-06-23	220000.00
9	Luisa	Martínez	11	2025-06-22	3200000.00
9	Luisa	Martínez	12	2025-06-21	1450000.00
10	Juan	Torres	13	2025-06-23	120000.00
10	Juan	Torres	14	2025-06-22	2750000.00
10	Juan	Torres	15	2025-06-21	380000.00

15 rows in set (0.00 sec)

### 5. Obtener el tipo de producto y los productos asociados con INNER JOIN .

```
mysql> SELECT
->     tp.descripcion AS tipo_producto,
->     p.nombre AS nombre_producto,
->     p.precio
-> FROM productos p
-> INNER JOIN tipos_productos tp ON p.id_tipo_producto = tp.id;
```

tipo_producto	nombre_producto	precio
Computadores	Laptop Lenovo Ideapad 3	3200000.00
Mouses	Mouse Logitech M185	65000.00
Smartphones	Smartphone Samsung A34	1450000.00
Cargadores	Cargador Samsung 25W	120000.00
Refrigeradoras	Refrigeradora LG 380L	2750000.00
Microondas	Microondas Haceb 20L	380000.00
Sofás	Sofá en L Gris Oscuro	2100000.00
Lámparas	Lámpara LED Escritorio	89000.00
Camisetas	Camiseta Nike Dri-FIT	95000.00
Zapatos	Zapatos Adidas Runfalcon	220000.00

10 rows in set (0.00 sec)

6. Listar todos los clientes y el número de pedidos realizados con COUNT y GROUP BY .

```
mysql> SELECT
->     dc.nombre,
->     dc.apellido,
->     COUNT(p.id) AS total_pedidos
-> FROM clientes c
-> LEFT JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> LEFT JOIN pedidos p ON c.id = p.id_cliente
-> GROUP BY c.id, dc.nombre, dc.apellido;
```

nombre	apellido	total_pedidos
María	Gómez	3
Jorge	Ríos	3
Ana	Castro	3
Luis	Fernández	3
Paola	Mejía	3

5 rows in set (0.00 sec)



7. Combinar Pedidos y Empleados para mostrar qué empleados gestionaron pedidos específicos.

```
mysql> SELECT
->     p.id AS id_pedido,
->     p.fecha_pedido,
->     de.nombre AS nombre_empleado,
->     de.apellido AS apellido_empleado
-> FROM pedidos p
-> INNER JOIN empleados e ON p.id_empleado = e.id
-> INNER JOIN datos_empleados de ON e.id_datos_empleado = de.id;
```

id_pedido	fecha_pedido	nombre_empleado	apellido_empleado
1	2025-06-23	Carlos	Ramírez
2	2025-06-22	Carlos	Ramírez
3	2025-06-21	Carlos	Ramírez
4	2025-06-23	María	López
5	2025-06-22	María	López
6	2025-06-21	María	López
7	2025-06-23	Andrés	González
8	2025-06-22	Andrés	González
9	2025-06-21	Andrés	González
10	2025-06-23	Luisa	Martínez
11	2025-06-22	Luisa	Martínez
12	2025-06-21	Luisa	Martínez
13	2025-06-23	Juan	Torres
14	2025-06-22	Juan	Torres
15	2025-06-21	Juan	Torres

8. Mostrar productos que no han sido pedidos ( RIGHT JOIN ).

```
mysql> SELECT
->     p.id AS id_producto,
->     p.nombre AS nombre_producto,
->     p.precio
-> FROM detalles_pedido dp
-> RIGHT JOIN productos p ON dp.id_producto = p.id
-> WHERE dp.id IS NULL;
```

id_producto	nombre_producto	precio
31	Monitor LG 24" Full HD	680000.00

```
|          32 | Auriculares Sony WH-CH520 | 320000.00 |
|          33 | Nevera Haceb No Frost 300L | 2150000.00 |
|          34 | Silla Ergonómica Ejecutiva | 420000.00 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

8. Mostrar el total de pedidos y ubicación de clientes usando múltiples JOIN .

```
mysql> SELECT
->     dc.nombre,
->     dc.apellido,
->     ci.descripcion AS ciudad,
->     es.descripcion AS estado,
->     pa.descripcion AS pais,
->     COUNT(p.id) AS total_pedidos
-> FROM clientes c
-> LEFT JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> LEFT JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
-> LEFT JOIN ciudad ci ON eu.id_ciudad = ci.id
-> LEFT JOIN estado es ON ci.id_estado = es.id
-> LEFT JOIN pais pa ON es.id_pais = pa.id
-> LEFT JOIN pedidos p ON c.id = p.id_cliente
-> GROUP BY c.id, dc.nombre, dc.apellido, ci.descripcion, es.descripcion,
pa.descripcion;
+-----+-----+-----+-----+-----+-----+
| nombre | apellido | ciudad | estado | pais | total_pedidos |
+-----+-----+-----+-----+-----+-----+
| María | Gómez | Armenia | Quindío | Colombia | 3 |
| Jorge | Ríos | Cartagena | Bolívar | Colombia | 3 |
| Ana | Castro | Neiva | Huila | Colombia | 3 |
| Luis | Fernández | Ibagué | Tolima | Colombia | 3 |
| Paola | Mejía | Popayán | Cauca | Colombia | 3 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

10. Unir Proveedores , Productos , y TiposProductos para un listado completo de inventario.

```
mysql> SELECT
-> dp.nombre AS nombre_proveedor,
-> p.nombre AS nombre_producto,
-> p.precio,
-> tp.descripcion AS tipo_producto
-> FROM productos p
-> INNER JOIN proveedores pr ON p.id_proveedor = pr.id
-> INNER JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id
-> INNER JOIN tipos_productos tp ON p.id_tipo_producto = tp.id;
```

nombre_proveedor	nombre_producto	precio	tipo_producto
Distribuidora Andina S.A.S	Laptop Lenovo Ideapad 3	3200000.00	Computadores
Distribuidora Andina S.A.S	Mouse Logitech M185	65000.00	Mouses
Proveedores del Norte LTDA	Smartphone Samsung A34	1450000.00	Smartphones
Proveedores del Norte LTDA	Cargador Samsung 25W	120000.00	Cargadores
Suministros Globales S.A.	Refrigeradora LG 380L	2750000.00	Refrigeradoras
Suministros Globales S.A.	Microondas Haceb 20L	380000.00	Microondas
Importadora del Centro	Sofá en L Gris Oscuro	2100000.00	Sofás
Importadora del Centro	Lámpara LED Escritorio	89000.00	Lámparas
TechMarket Colombia	Camiseta Nike Dri-FIT	95000.00	Camisetas
TechMarket Colombia	Zapatos Adidas Runfalcon	220000.00	Zapatos

10 rows in set (0.00 sec)

## Consultas Simples

1. Seleccionar todos los productos con precio mayor a \$500000.

```
mysql> SELECT * FROM productos WHERE precio >500000;
```

id	nombre	precio	id_tipo_producto	id_proveedor
1	Laptop Lenovo Ideapad 3	3200000.00	1	17
3	Smartphone Samsung A34	1450000.00	2	18
5	Refrigeradora LG 380L	2750000.00	4	19
7	Sofá en L Gris Oscuro	2100000.00	7	20
31	Monitor LG 24" Full HD	680000.00	1	17
33	Nevera Haceb No Frost 300L	2150000.00	4	19

6 rows in set (0.00 sec)

2. Consultar clientes registrados en una ciudad específica.

```
mysql> SELECT dc.nombre, dc.apellido, c.descripcion AS ciudad
-> FROM clientes cl
-> JOIN datos_cliente dc ON cl.id_datos_cliente = dc.id
-> JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
-> JOIN ciudad c ON eu.id_ciudad = c.id
-> WHERE c.descripcion = 'Cartagena';
```

nombre	apellido	ciudad
Jorge	Ríos	Cartagena

1 row in set (0.00 sec)

3. Mostrar empleados contratados en los últimos 2 años.

```
mysql> SELECT
-> e.id AS id_Empleado,
-> de.nombre,
-> de.apellido,
-> e.fecha_ingreso,
-> ca.descripcion AS cargo
-> FROM empleados e
-> JOIN datos_empleados de ON e.id_datos_Empleado = de.id
-> JOIN cargo ca ON e.id_cargo = ca.id
-> WHERE e.fecha_ingreso >= DATE_SUB(CURDATE(), INTERVAL 2 YEAR);
```

id_Empleado	nombre	apellido	fecha_ingreso	cargo
12	Carlos	Ramírez	2024-01-12	Vendedor
16	Carlos	Ramírez	2025-11-10	Vendedor

4. Seleccionar proveedores que suministran más de 5 productos.

```
mysql> SELECT pr.id, dp.nombre, COUNT(p.id) AS total_productos
-> FROM proveedores pr
-> JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id
-> JOIN productos p ON p.id_proveedor = pr.id
-> GROUP BY pr.id, dp.nombre
-> HAVING total_productos > 5;
```

id	nombre	total_productos
23	MegaProveedor	6

1 row in set (0.00 sec)

5. Listar clientes que no tienen dirección registrada en UbicacionCliente .

```
mysql> SELECT dc.nombre, dc.apellido
-> FROM clientes c
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> WHERE dc.id_direccion_ubicacion IS NULL;
```

Empty set (0.00 sec)

6. Calcular el total de ventas por cada cliente.

```
mysql> SELECT dc.nombre, dc.apellido, SUM(p.total) AS total_compras
-> FROM pedidos p
-> JOIN clientes c ON p.id_cliente = c.id
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> GROUP BY c.id, dc.nombre, dc.apellido;
```

nombre	apellido	total_compras
--------	----------	---------------

nombre	apellido	total_compras
María	Gómez	4715000.00
Jorge	Ríos	3250000.00
Ana	Castro	2284000.00
Luis	Fernández	4870000.00
Paola	Mejía	3250000.00

5 rows in set (0.00 sec)

7. Mostrar el salario promedio de los empleados.

```
mysql> SELECT AVG(salario) AS salario_promedio
-> FROM empleados
-> WHERE salario IS NOT NULL;
```

salario_promedio
2875000.000000

1 row in set (0.00 sec)

8. Consultar el tipo de productos disponibles en TiposProductos .

```
mysql> SELECT id, descripcion
-> FROM tipos_productos LIMIT 10;
```

id	descripcion
1	Computadores
2	Smartphones
3	Tablets
4	Refrigeradoras
5	Lavadoras
6	Microondas
7	Sofás
8	Lámparas
9	Cuadros_Decorativos
10	Camisetas

10 rows in set (0.00 sec)

9. Seleccionar los 3 productos más caros.

```
mysql> SELECT nombre, precio
-> FROM productos
-> ORDER BY precio DESC
```

```
-> LIMIT 3;
```

```
+-----+-----+
| nombre                | precio    |
+-----+-----+
| Laptop Lenovo Ideapad 3 | 3200000.00 |
| Refrigeradora LG 380L   | 2750000.00 |
| Nevera Haceb No Frost 300L | 2150000.00 |
+-----+-----+
3 rows in set (0.00 sec)
```

10. Consultar el cliente con el mayor número de pedidos.

```
mysql> SELECT dc.nombre, dc.apellido, COUNT(p.id) AS cantidad_pedidos
-> FROM pedidos p
-> JOIN clientes c ON p.id_cliente = c.id
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> GROUP BY c.id, dc.nombre, dc.apellido
-> ORDER BY cantidad_pedidos DESC
-> LIMIT 1;
```

```
+-----+-----+-----+
| nombre | apellido | cantidad_pedidos |
+-----+-----+-----+
| María  | Gómez    | 3                |
+-----+-----+-----+
1 row in set (0.00 sec)
```

## Consultas Multitabla

1. Listar todos los pedidos y el cliente asociado.

```
mysql> SELECT
->     p.id AS id_pedido,
->     dc.nombre AS nombre_cliente,
->     dc.apellido AS apellido_cliente,
->     p.fecha_pedido,
->     p.total
-> FROM
->     pedidos p
->     JOIN clientes c ON p.id_cliente = c.id
->     JOIN datos_cliente dc ON c.id_datos_cliente = dc.id;
```

```
+-----+-----+-----+-----+-----+
| id_pedido | nombre_cliente | apellido_cliente | fecha_pedido | total    |
+-----+-----+-----+-----+-----+
| 1         | María         | Gómez           | 2025-06-23   | 3200000.00 |
+-----+-----+-----+-----+-----+
```

2	María	Gómez	2025-06-22	65000.00
3	María	Gómez	2025-06-21	1450000.00
4	Jorge	Ríos	2025-06-23	120000.00
5	Jorge	Ríos	2025-06-22	2750000.00
6	Jorge	Ríos	2025-06-21	380000.00
7	Ana	Castro	2025-06-23	2100000.00
8	Ana	Castro	2025-06-22	89000.00
9	Ana	Castro	2025-06-21	95000.00
10	Luis	Fernández	2025-06-23	220000.00
11	Luis	Fernández	2025-06-22	3200000.00
12	Luis	Fernández	2025-06-21	1450000.00
13	Paola	Mejía	2025-06-23	120000.00
14	Paola	Mejía	2025-06-22	2750000.00
15	Paola	Mejía	2025-06-21	380000.00

15 rows in set (0.00 sec)

2. Mostrar la ubicación de cada cliente en sus pedidos.

```
mysql> SELECT
->     p.id AS id_pedido,
->     dc.nombre,
->     dc.apellido,
->     ci.descripcion AS ciudad,
->     d.descripcion AS direccion
-> FROM
->     pedidos p
->     JOIN clientes c ON p.id_cliente = c.id
->     JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
->     JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
->     JOIN ciudad ci ON eu.id_ciudad = ci.id
->     JOIN direcciones d ON eu.id_direccion = d.id;
```

ERROR 1054 (42S22): Unknown column 'd.descripcion' in 'field list'

```
mysql> DESC direcciones;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
direccion	varchar(255)	YES		NULL	
codigo_postal	varchar(10)	YES		NULL	

3 rows in set (0.01 sec)

### 3. Listar productos junto con el proveedor y tipo de producto.

```
mysql> SELECT
->     p.nombre AS producto,
->     tp.descripcion AS tipo_producto,
->     dp.nombre AS proveedor
-> FROM
->     productos p
->     JOIN tipos_productos tp ON p.id_tipo_producto = tp.id
->     JOIN proveedores pr ON p.id_proveedor = pr.id
->     JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id;
```

producto	tipo_producto	proveedor
Laptop Lenovo Ideapad 3	Computadores	Distribuidora Andina S.A.S
Mouse Logitech M185	Mouses	Distribuidora Andina S.A.S
Monitor LG 24" Full HD	Computadores	Distribuidora Andina S.A.S
Smartphone Samsung A34	Smartphones	Proveedores del Norte LTDA
Cargador Samsung 25W	Cargadores	Proveedores del Norte LTDA
Auriculares Sony WH-CH520	Smartphones	Proveedores del Norte LTDA
Refrigeradora LG 380L	Refrigeradoras	Suministros Globales S.A.
Microondas Haceb 20L	Microondas	Suministros Globales S.A.
Nevera Haceb No Frost 300L	Refrigeradoras	Suministros Globales S.A.
Sofá en L Gris Oscuro	Sofás	Importadora del Centro
Lámpara LED Escritorio	Lámparas	Importadora del Centro
Silla Ergonómica Ejecutiva	Lámparas	Importadora del Centro
Camiseta Nike Dri-FIT	Camisetas	TechMarket Colombia
Zapatos Adidas Runfalcon	Zapatos	TechMarket Colombia
Producto A	Computadores	MegaProveedor
Producto B	Computadores	MegaProveedor
Producto C	Computadores	MegaProveedor
Producto D	Computadores	MegaProveedor
Producto E	Computadores	MegaProveedor
Producto F	Computadores	MegaProveedor

20 rows in set (0.00 sec)

### 4. Consultar todos los empleados que gestionan pedidos de clientes en una ciudad específica.

```
mysql> SELECT DISTINCT
->     de.nombre,
->     de.apellido,
->     ci.descripcion AS ciudad
```



```

-> FROM
->     pedidos p
->     JOIN empleados e ON p.id_empleado = e.id
->     JOIN datos_empleados de ON e.id_datos_empleado = de.id
->     JOIN clientes c ON p.id_cliente = c.id
->     JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
->     JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
->     JOIN ciudad ci ON eu.id_ciudad = ci.id
-> WHERE
->     ci.descripcion = 'Bogotá';
Empty set (0.00 sec)

```

##### 5. Consultar los 5 productos más vendidos.

```

mysql> SELECT
->     p.nombre,
->     SUM(dp.cantidad) AS total_vendido
-> FROM
->     detalles_pedido dp
->     JOIN productos p ON dp.id_producto = p.id
-> GROUP BY
->     p.nombre
-> ORDER BY
->     total_vendido DESC
-> LIMIT 5;

```

nombre	total_vendido
Cargador Samsung 25W	4
Zapatos Adidas Runfalcon	3
Refrigeradora LG 380L	3
Laptop Lenovo Ideapad 3	2
Microondas Haceb 20L	2

5 rows in set (0.00 sec)

##### 6. Obtener la cantidad total de pedidos por cliente y ciudad.

```

mysql> SELECT
->     dc.nombre,
->     dc.apellido,
->     ci.descripcion AS ciudad,
->     COUNT(p.id) AS total_pedidos
-> FROM

```

```

-> pedidos p
-> JOIN clientes c ON p.id_cliente = c.id
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
-> JOIN ciudad ci ON eu.id_ciudad = ci.id
-> GROUP BY
-> dc.nombre, dc.apellido, ci.descripcion;
+-----+-----+-----+-----+
| nombre | apellido | ciudad | total_pedidos |
+-----+-----+-----+-----+
| María  | Gómez    | Armenia | 3 |
| Jorge  | Ríos     | Cartagena | 3 |
| Ana    | Castro    | Neiva    | 3 |
| Luis   | Fernández | Ibagué   | 3 |
| Paola  | Mejía     | Popayán  | 3 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

7. Listar clientes y proveedores en la misma ciudad.

```

mysql> SELECT
-> dc.nombre AS cliente,
-> dp.nombre AS proveedor,
-> ci.descripcion AS ciudad
-> FROM
-> datos_cliente dc
-> JOIN entidad_ubicacion ecu ON dc.id_direccion_ubicacion = ecu.id
-> JOIN ciudad ci ON ecu.id_ciudad = ci.id
-> JOIN datos_proveedor dp ON dp.id_direccion_ubicacion IN (
-> SELECT id FROM entidad_ubicacion WHERE id_ciudad = ci.id
-> );
+-----+-----+-----+
| cliente | proveedor | ciudad |
+-----+-----+-----+
| Carlos  | MegaProveedor | Leticia |
+-----+-----+-----+
1 row in set (0.00 sec)

```

8. Mostrar el total de ventas agrupado por tipo de producto.

```

mysql> SELECT
-> tp.descripcion AS tipo_producto,
-> SUM(dp.cantidad * p.precio) AS total_ventas
-> FROM

```

```

->     detalles_pedido dp
->     JOIN productos p ON dp.id_producto = p.id
->     JOIN tipos_productos tp ON p.id_tipo_producto = tp.id
-> GROUP BY
->     tp.descripcion;
+-----+-----+
| tipo_producto | total_ventas |
+-----+-----+
| Computadores  | 6400000.00 |
| Smartphones   | 1450000.00 |
| Refrigeradoras | 8250000.00 |
| Microondas    | 760000.00  |
| Mouses        | 65000.00   |
| Cargadores    | 480000.00  |
| Sofás         | 2100000.00 |
| Lámparas      | 178000.00  |
| Camisetas     | 190000.00  |
| Zapatos       | 660000.00  |
+-----+-----+
10 rows in set (0.00 sec)

```

9. Listar empleados que gestionan pedidos de productos de un proveedor específico.

```

mysql> SELECT DISTINCT
->     de.nombre,
->     de.apellido
-> FROM
->     pedidos pe
->     JOIN empleados e ON pe.id_empleado = e.id
->     JOIN datos_empleados de ON e.id_datos_empleado = de.id
->     JOIN detalles_pedido dp ON dp.id_pedido = pe.id
->     JOIN productos p ON dp.id_producto = p.id
-> WHERE
->     p.id_proveedor = 17;
+-----+-----+
| nombre | apellido |
+-----+-----+
| Carlos | Ramírez  |
| María  | López    |
+-----+-----+
2 rows in set (0.00 sec)

```

10. Obtener el ingreso total de cada proveedor a partir de los productos vendidos.

```
mysql> SELECT
->     dp.nombre AS proveedor,
->     SUM(dpv.cantidad * p.precio) AS ingreso_total
-> FROM
->     productos p
->     JOIN proveedores pr ON p.id_proveedor = pr.id
->     JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id
->     JOIN detalles_pedido dpv ON dpv.id_producto = p.id
-> GROUP BY
->     dp.nombre;
```

proveedor	ingreso_total
Distribuidora Andina S.A.S	6465000.00
Proveedores del Norte LTDA	1930000.00
Suministros Globales S.A.	9010000.00
Importadora del Centro	2278000.00
TechMarket Colombia	850000.00

5 rows in set (0.00 sec)

## Subconsultas

1. Consultar el producto más caro en cada categoría.

```
mysql> SELECT tp.id_categoria, cp.descripcion AS categoría, p.nombre, p.precio
-> FROM productos p
-> JOIN tipos_productos tp ON p.id_tipo_producto = tp.id
-> JOIN categorias_p cp ON tp.id_categoria = cp.id
-> WHERE p.precio = (
->     SELECT MAX(p2.precio)
->     FROM productos p2
->     JOIN tipos_productos tp2 ON p2.id = tp2.id
->     WHERE tp2.id_categoria = tp.id_categoria
-> );
```

id_categoria	categoría	nombre	precio
1	Tecnología	Laptop Lenovo Ideapad 3	3200000.00
2	Electrodomésticos	Refrigeradora LG 380L	2750000.00
3	Hogar_Decoración	Sofá en L Gris Oscuro	2100000.00
4	Ropa_Accesorios	Zapatos Adidas Runfalcon	220000.00

4 rows in set (0.05 sec)

2. Encontrar el cliente con mayor total en pedidos.

```
mysql> SELECT
->   c.id, dc.nombre, dc.apellido, sq.total_pedidos
-> FROM clientes AS c
-> JOIN datos_cliente AS dc ON c.id_datos_cliente = dc.id
-> JOIN (
->   SELECT p.id_cliente, SUM(p.total) AS total_pedidos
->   FROM pedidos AS p
->   GROUP BY
->     p.id_cliente
->   ORDER BY
->     total_pedidos DESC
->   LIMIT 1
-> ) AS sq ON c.id = sq.id_cliente;
```

id	nombre	apellido	total_pedidos
9	Luis	Fernández	4870000.00

1 row in set (0.01 sec)

3. Listar empleados que ganan más que el salario promedio.

```
mysql> SELECT e.id, de.nombre, de.apellido, e.salario
-> FROM empleados e
-> JOIN datos_empleados de ON e.id_datos_empleado = de.id
-> WHERE e.salario > (
->   SELECT AVG(salario) FROM empleados
-> );
```

id	nombre	apellido	salario
7	María	López	3200000.00
10	Juan	Torres	3200000.00
12	Carlos	Ramírez	3200000.00

3 rows in set (0.00 sec)

4. Consultar productos que han sido pedidos más de 5 veces.

```
mysql> SELECT p.id, p.nombre, COUNT(dp.id) AS veces_pedido
-> FROM productos p
-> JOIN detalles_pedido dp ON p.id = dp.id_producto
-> GROUP BY p.id
-> HAVING COUNT(dp.id) > 5;
```

Empty set (0.03 sec)

5. Listar pedidos cuyo total es mayor al promedio de todos los pedidos.

```
mysql> SELECT
->     p.id AS id_pedido,
->     p.fecha_pedido,
->     p.total,
->     dc.nombre AS nombre_cliente,
->     dc.apellido AS apellido_cliente
-> FROM
->     pedidos p
-> JOIN
->     clientes c ON p.id_cliente = c.id
-> JOIN
->     datos_cliente dc ON c.id_datos_cliente = dc.id
-> WHERE
->     p.total > (SELECT AVG(total) FROM pedidos);
```

id_pedido	fecha_pedido	total	nombre_cliente	apellido_cliente
1	2025-06-23	3200000.00	María	Gómez
3	2025-06-21	1450000.00	María	Gómez
5	2025-06-22	2750000.00	Jorge	Ríos
7	2025-06-23	2100000.00	Ana	Castro
11	2025-06-22	3200000.00	Luis	Fernández
12	2025-06-21	1450000.00	Luis	Fernández
14	2025-06-22	2750000.00	Paola	Mejía

7 rows in set (0.00 sec)

4. Seleccionar los 3 proveedores con más productos.

```
mysql> SELECT pr.id, dp.nombre, COUNT(p.id) AS total_productos
-> FROM proveedores pr
-> JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id
-> JOIN productos p ON p.id_proveedor = pr.id
-> GROUP BY pr.id
-> ORDER BY total_productos DESC
-> LIMIT 3;
```

id	nombre	total_productos
23	MegaProveedor	6
17	Distribuidora Andina S.A.S	3
18	Proveedores del Norte LTDA	3

3 rows in set (0.00 sec)

7. Consultar productos con precio superior al promedio en su tipo.

```
mysql> SELECT p.*
-> FROM productos p
-> WHERE p.precio > (
->     SELECT AVG(p2.precio)
->     FROM productos p2
->     WHERE p2.id_tipo_producto = p.id_tipo_producto
-> );
```

id	nombre	precio	id_tipo_producto	id_proveedor
1	Laptop Lenovo Ideapad 3	3200000.00	1	17
3	Smartphone Samsung A34	1450000.00	2	18
5	Refrigeradora LG 380L	2750000.00	4	19
31	Monitor LG 24" Full HD	680000.00	1	17
34	Silla Ergonómica Ejecutiva	420000.00	8	20

5 rows in set (0.00 sec)

8. Mostrar clientes que han realizado más pedidos que la media.

```
mysql> SELECT c.id, dc.nombre, COUNT(p.id) AS pedidos
-> FROM clientes c
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id
-> JOIN pedidos p ON p.id_cliente = c.id
-> GROUP BY c.id
-> HAVING COUNT(p.id) > (
->   SELECT AVG(num_pedidos)
->   FROM (
->     SELECT COUNT(*) AS num_pedidos
->     FROM pedidos
->     GROUP BY id_cliente
->   ) AS sub
-> );
Empty set (0.00 sec)
```

9. Encontrar productos cuyo precio es mayor que el promedio de todos los productos.

```
mysql> SELECT
->   p.id AS id_producto,
->   p.nombre AS nombre_producto,
->   p.precio,
->   tp.descripcion AS tipo_producto,
->   cat.descripcion AS categoria_producto
-> FROM
->   productos p
-> JOIN
->   tipos_productos tp ON p.id_tipo_producto = tp.id
-> JOIN
->   categorias_p cat ON tp.id_categoria = cat.id
-> WHERE
->   p.precio > (SELECT AVG(precio) FROM productos);
```

id_producto	nombre_producto	precio	tipo_producto	categoria_producto
1	Laptop Lenovo Ideapad 3	3200000.00	Computadores	Tecnología
3	Smartphone Samsung A34	1450000.00	Smartphones	Tecnología
5	Refrigeradora LG 380L	2750000.00	Refrigeradoras	Electrodomésticos
7	Sofá en L Gris Oscuro	2100000.00	Sofás	Hogar_Decoración
33	Nevera Haceb No Frost 300L	2150000.00	Refrigeradoras	Electrodomésticos

5 rows in set (0.00 sec)

10. Mostrar empleados cuyo salario es menor al promedio del departamento.

```
mysql> SELECT e.id, de.nombre, de.apellido, e.salario, ca.descripcion AS cargo
-> FROM empleados e
-> JOIN datos_empleados de ON e.id_datos_empleado = de.id
-> JOIN cargo ca ON e.id_cargo = ca.id
-> WHERE e.salario < (
->   SELECT AVG(e2.salario)
->   FROM empleados e2
->   WHERE e2.id_cargo = e.id_cargo
-> )LIMIT 1;
```

id	nombre	apellido	salario	cargo
6	Carlos	Ramírez	2500000.00	Vendedor

1 row in set (0.00 sec)

## Procedimientos Almacenados

1. Crear un procedimiento para actualizar el precio de todos los productos de un proveedor.

```
DELIMITER $$
```

```
CREATE PROCEDURE actualizar_precio_proveedor(  
    IN proveedor_id INT,  
    IN porcentaje DECIMAL(5,2)  
)  
BEGIN  
    UPDATE productos  
    SET precio = precio * (1 + porcentaje / 100)  
    WHERE id_proveedor = proveedor_id;  
END$$
```

```
DELIMITER ;
```

```
mysql> SELECT id, nombre, precio FROM productos WHERE id_proveedor = 17;
```

id	nombre	precio
1	Laptop Lenovo Ideapad 3	3291200.00
2	Mouse Logitech M185	78650.00
31	Monitor LG 24" Full HD	699380.00

```
3 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql>
```

```
mysql> CALL actualizar_precio_proveedor(17, 10);
```

```
Query OK, 3 rows affected (0.00 sec)
```

```
mysql>
```

```
mysql>
```

```
mysql> SELECT id, nombre, precio FROM productos WHERE id_proveedor = 17;
```

id	nombre	precio
1	Laptop Lenovo Ideapad 3	3620320.00
2	Mouse Logitech M185	86515.00
31	Monitor LG 24" Full HD	769318.00

```
3 rows in set (0.00 sec)
```

2. Un procedimiento que devuelva la dirección de un cliente por ID.

```
DELIMITER $$
```

```
CREATE PROCEDURE obtener_direccion_cliente(  
    IN cliente_id INT  
)  
BEGIN  
    SELECT d.direccion, d.codigo_postal, c.descripcion AS ciudad, e.descripcion  
    AS estado, p.descripcion AS pais  
    FROM clientes cl  
    JOIN datos_cliente dc ON cl.id_datos_cliente = dc.id
```



```

JOIN entidad_ubicacion eu ON dc.id_direccion_ubicacion = eu.id
JOIN direcciones d ON eu.id_direccion = d.id
JOIN ciudad c ON eu.id_ciudad = c.id
JOIN estado e ON c.id_estado = e.id
JOIN pais p ON e.id_pais = p.id
WHERE cl.id = cliente_id;
END$$

```

```

DELIMITER ;

```

```

mysql> CALL obtener_direccion_cliente(6);
+-----+-----+-----+-----+-----+
| direccion | codigo_postal | ciudad | estado | pais |
+-----+-----+-----+-----+-----+
| Carrera 50 #60-20 | 110711 | Armenia | Quindío | Colombia |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

```

3. Crear un procedimiento que registre un pedido nuevo y sus detalles.

```

DELIMITER $$

```

```

CREATE PROCEDURE registrar_pedido(
    IN cliente_id INT,
    IN empleado_id INT,
    IN fecha DATE,
    IN producto_id INT,
    IN cantidad INT,
    IN precio_unitario DECIMAL(10,2)
)
BEGIN
    DECLARE nuevo_pedido_id INT;
    DECLARE total DECIMAL(10,2);

    SET total = cantidad * precio_unitario;

    INSERT INTO pedidos (id_cliente, id_empleado, fecha_pedido, total)
    VALUES (cliente_id, empleado_id, fecha, total);

    SET nuevo_pedido_id = LAST_INSERT_ID();

    INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad, precio_unit)
    VALUES (nuevo_pedido_id, producto_id, cantidad, precio_unitario);
END$$

```

```
DELIMITER ;
```

```
mysql> CALL registrar_pedido(6, 6, CURDATE(), 1, 3, 50000);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT  
-> p.id AS id_pedido,  
-> dc.nombre AS nombre_cliente,  
-> dc.apellido AS apellido_cliente,  
-> de.nombre AS nombre_empleado,  
-> de.apellido AS apellido_empleado,  
-> p.fecha_pedido,  
-> p.total  
-> FROM pedidos p  
-> JOIN clientes c ON p.id_cliente = c.id  
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id  
-> JOIN empleados e ON p.id_empleado = e.id  
-> JOIN datos_empleados de ON e.id_datos_empleado = de.id  
-> ORDER BY p.id DESC  
-> LIMIT 1;
```

id_pedido	nombre_cliente	apellido_cliente	nombre_empleado	apellido_empleado	fecha_pedido	total
19	María	Gómez	Carlos	Ramírez	2025-07-04	150000.00

1 row in set (0.00 sec)

```
mysql>  
mysql> SELECT  
-> p.id AS id_pedido,  
-> dc.nombre AS nombre_cliente,  
-> de.nombre AS nombre_empleado,  
-> pr.nombre AS producto,  
-> dp.cantidad,  
-> dp.precio_unit,  
-> (dp.cantidad * dp.precio_unit) AS subtotal,  
-> p.total AS total_pedido,  
-> p.fecha_pedido  
-> FROM pedidos p  
-> JOIN clientes c ON p.id_cliente = c.id  
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id  
-> JOIN empleados e ON p.id_empleado = e.id  
-> JOIN datos_empleados de ON e.id_datos_empleado = de.id  
-> JOIN detalles_pedido dp ON dp.id_pedido = p.id  
-> JOIN productos pr ON dp.id_producto = pr.id  
-> ORDER BY p.id DESC  
-> LIMIT 1;
```

id_pedido	nombre_cliente	nombre_empleado	producto	cantidad	precio_unit	subtotal	total_pedido	fecha_pedido
19	María	Carlos	Laptop Lenovo Ideapad 3	3	50000.00	150000.00	150000.00	2025-07-04

1 row in set (0.00 sec)

4. Un procedimiento para calcular el total de ventas de un cliente.

```
DELIMITER $$
```

```
CREATE PROCEDURE total_ventas_cliente(  
  IN cliente_id INT  
)  
BEGIN  
  SELECT SUM(total) AS total_ventas  
  FROM pedidos  
  WHERE id_cliente = cliente_id;  
END$$
```

```
DELIMITER ;
```

```
mysql> SELECT c.id AS id_cliente, dc.nombre, dc.apellido
-> FROM clientes c
-> JOIN datos_cliente dc ON c.id_datos_cliente = dc.id;
```

id_cliente	nombre	apellido
6	María	Gómez
7	Jorge	Ríos
8	Ana	Castro
9	Luis	Fernández
10	Paola	Mejía
16	Carlos	Ramírez

6 rows in set (0.00 sec)

```
mysql> CALL total_ventas_cliente(6);
```

total_ventas
5015000.00

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

5. Crear un procedimiento para obtener los empleados por puesto.

```
DELIMITER $$

CREATE PROCEDURE empleados_por_puesto(
    IN id_cargo INT
)
BEGIN
    SELECT e.id, de.nombre, de.apellido
    FROM empleados e
    JOIN datos_empleados de ON e.id_datos_empleado = de.id
    WHERE e.id_cargo = id_cargo;
END$$

DELIMITER ;
```

```
mysql> SELECT id, descripcion FROM cargo;
```

id	descripcion
1	Vendedor
2	Administrador
3	Gerente
4	Asistente
5	Supervisor
6	Desarrollador

6 rows in set (0.00 sec)

```
mysql> CALL empleados_por_puesto(5);
```

id	nombre	apellido
10	Juan	Torres

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

6. Un procedimiento que actualice el salario de empleados por puesto.

```

DELIMITER $$

CREATE PROCEDURE actualizar_salario_por_puesto(
    IN id_cargo INT,
    IN nuevo_salario DECIMAL(10,2)
)
BEGIN
    UPDATE empleados
    SET salario = nuevo_salario
    WHERE id_cargo = id_cargo;
END$$

DELIMITER ;

```

```

mysql> SELECT id, id_cargo, salario FROM empleados WHERE id_cargo = 2;
+----+-----+-----+
| id | id_cargo | salario |
+----+-----+-----+
| 7  | 2       | 3500000.00 |
+----+-----+-----+
1 row in set (0.00 sec)

mysql> CALL actualizar_salario_por_puesto(1, 5000000);
Query OK, 8 rows affected (0.01 sec)

mysql> SELECT id, id_cargo, salario FROM empleados WHERE id_cargo = 2;
+----+-----+-----+
| id | id_cargo | salario |
+----+-----+-----+
| 7  | 2       | 5000000.00 |
+----+-----+-----+
1 row in set (0.00 sec)

```

7. Crear un procedimiento que liste los pedidos entre dos fechas.

```

DELIMITER $$

CREATE PROCEDURE pedidos_entre_fechas(
    IN fecha_inicio DATE,
    IN fecha_fin DATE
)
BEGIN
    SELECT * FROM pedidos
    WHERE fecha_pedido BETWEEN fecha_inicio AND fecha_fin;
END$$

DELIMITER ;

```

```
mysql> CALL pedidos_entre_fechas('2024-01-01', '2025-12-31');
```

id	id_cliente	id_empleado	fecha_pedido	total
1	6	6	2025-06-23	3200000.00
2	6	6	2025-06-22	65000.00
3	6	6	2025-06-21	1450000.00
4	7	7	2025-06-23	120000.00
5	7	7	2025-06-22	2750000.00
6	7	7	2025-06-21	380000.00
7	8	8	2025-06-23	2100000.00
8	8	8	2025-06-22	89000.00
9	8	8	2025-06-21	95000.00
10	9	9	2025-06-23	220000.00
11	9	9	2025-06-22	3200000.00
12	9	9	2025-06-21	1450000.00
13	10	10	2025-06-23	120000.00
14	10	10	2025-06-22	2750000.00
15	10	10	2025-06-21	380000.00

```
15 rows in set (0.00 sec)

Query OK, 0 rows affected (0.07 sec)
```

8. Un procedimiento para aplicar un descuento a productos de una categoría.

```
DELIMITER $$
```

```
CREATE PROCEDURE aplicar_descuento_categoria(
    IN id_categoria INT,
    IN porcentaje_descuento DECIMAL(5,2)
)
BEGIN
    UPDATE productos p
    JOIN tipos_productos tp ON p.id_tipo_producto = tp.id
    SET p.precio = p.precio * (1 - porcentaje_descuento / 100)
    WHERE tp.id_categoria = id_categoria;
END$$
```

```
DELIMITER ;
```

```
mysql> SELECT p.nombre, p.precio
-> FROM productos p
-> JOIN tipos_productos tp ON p.id_tipo_producto = tp.id
-> WHERE tp.id_categoria = 1;
```

nombre	precio
Laptop Lenovo Ideapad 3	2720000.00
Monitor LG 24" Full HD	578000.00
Producto A	8500.00
Producto B	17000.00
Producto C	25500.00
Producto D	34000.00
Producto E	42500.00
Producto F	51000.00
Smartphone Samsung A34	1232500.00
Auriculares Sony WH-CH520	272000.00

```
10 rows in set (0.00 sec)
```

9. Crear un procedimiento que liste todos los proveedores de un tipo de producto.

```
DELIMITER $$

CREATE PROCEDURE proveedores_por_tipo_producto(
    IN tipo_producto_id INT
)
BEGIN
    SELECT DISTINCT pr.id, dp.nombre
    FROM productos p
    JOIN proveedores pr ON p.id_proveedor = pr.id
    JOIN datos_proveedor dp ON pr.id_datos_proveedor = dp.id
    WHERE p.id_tipo_producto = tipo_producto_id;
END$$

DELIMITER ;
```

```
mysql> CALL proveedores_por_tipo_producto(1);
```

id	nombre
17	Distribuidora Andina S.A.S
23	MegaProveedor

```
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

10. Un procedimiento que devuelva el pedido de mayor valor.

```
DELIMITER $$

CREATE PROCEDURE pedido_mayor_valor()
BEGIN
```

```
SELECT * FROM pedidos
ORDER BY total DESC
LIMIT 1;
END$$
```

```
DELIMITER ;
```

```
mysql> CALL pedido_mayor_valor();
```

id	id_cliente	id_employado	fecha_pedido	total
1	6	6	2025-06-23	3200000.00

```
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```