

**Universidad Nacional Autónoma de México**

Facultad de Ingeniería

Ingeniería en Computación

Proyecto final de la asignatura de bases de datos

Profesor: Ing. Fernando Arreola

18 de Noviembre de 2024

Alumnos:

Aquino Lozada Gabriela

Blanco Pulido Gabriel Alonzo

Hernández Pérez Yair Edwin

Rosas Cañada Abraham

Sánchez Gachuz Jenyfer Estefanya

```

-- tabla PROVEEDOR
/*
Almacena informacion basica de los proveedores identificamos el origen de cada producto
*/
CREATE TABLE PROVEEDOR (
    id_proveedor INT DEFAULT nextval('seq_proveedor'),--id unico autoincremental
    razon_social VARCHAR(100) NOT NULL,--nombre proveedor
    telefono VARCHAR(15) NOT NULL,--telefono de contacto
    estado VARCHAR(50) NOT NULL,--estado del proveedor
    codigo_postal VARCHAR(5) NOT NULL,--codigo postal
    colonia VARCHAR(50) NOT NULL,--colonia
    calle VARCHAR(100) NOT NULL,--calle
    numero VARCHAR(10) NOT NULL,--numero exterior
    CONSTRAINT pk_proveedor PRIMARY KEY (id_proveedor),--llave primaria
    CONSTRAINT uk_proveedor_razon_social UNIQUE (razon_social)--razon social unica
);
--tabla CLIENTE

```

```

--tabla CLIENTE

/*
Almacena informacion de los clientes
permite asociar ventas con clientes y calcular ingresos del cliente
*/
CREATE TABLE CLIENTE (
    id_cliente INT DEFAULT nextval('seq_cliente'),--id unico autoincremental
    rfc VARCHAR(13) NOT NULL,--rfc del cliente
    nombre VARCHAR(50) NOT NULL,--nombre del cliente
    ap_paterno VARCHAR(50) NOT NULL,--apellido paterno
    ap_materno VARCHAR(50),--apellido materno opcional
    email VARCHAR(100) NOT NULL,--correo del cliente
    estado VARCHAR(50) NOT NULL,--estado del cliente
    codigo_postal VARCHAR(5) NOT NULL,-- código postal
    colonia VARCHAR(50) NOT NULL,--colonia
    calle VARCHAR(100) NOT NULL,--calle
    numero VARCHAR(10) NOT NULL,--numero exterior
    CONSTRAINT pk_cliente PRIMARY KEY (id_cliente),--llave primaria
    CONSTRAINT uk_cliente_rfc UNIQUE (rfc),--rfc unico
    CONSTRAINT uk_cliente_email UNIQUE (email)--email unico
);

```

```
-- tabla EMPLEADO
/*
almacena informacion de los empleados que realizan ventas
asociar las ventas al empleado permitiendo calcular ordenes registradas por cada uno
despues relacion con dasnoard para empleados con más ordenes
*/

CREATE TABLE EMPLEADO (
  id_empleado INT DEFAULT nextval('seq_empleado'),--id unico auto incremental
  clave VARCHAR(10) NOT NULL,-- clave unica para identificar al empleado
  nombre VARCHAR(50) NOT NULL,--nombre del empleado
  ap_paterno VARCHAR(50) NOT NULL,--apellido
  ap_materno VARCHAR(50),--apellido materno opcional
  fecha_nacimiento DATE NOT NULL,--fecha de nacimiento del empleado
  fecha_ingreso DATE NOT NULL,-- fecha que ingreso al empleo
  CONSTRAINT pk_empleado PRIMARY KEY (id_empleado),--llave priamria
  CONSTRAINT uk_empleado_clave UNIQUE (clave),--clave unica
  CONSTRAINT chk_empleado_fecha_ingreso CHECK (fecha_ingreso >= fecha_nacimiento)
  --check para validar que la edad no tenga incoherencias
);
```

```
-- tabla PRODUCTO
/*
contiene datos del inventario para calcular ventas, utilidad y edentificar el top 3
*/

CREATE TABLE PRODUCTO (
  id_producto INT DEFAULT nextval('seq_producto'),--id unico autocrimental
  id_proveedor INT NOT NULL,--relacion con la tabla proveedor
  codigo_barras VARCHAR(13) NOT NULL,--codigo de barras unico
  precio_compra DECIMAL(10,2) NOT NULL,--precio de compra del producto en decimal
  foto_url VARCHAR(255),-- foto del producto en varchar para no subir fotos deberia ser blob
  fecha_compra DATE NOT NULL,--fecha de adquisicion
  stock INT NOT NULL,--cantidad disponible en el inventario
  tipo VARCHAR(20) NOT NULL,--tipo (papeleria, recarga, impresion , regalo)
  marca VARCHAR(50) NOT NULL,--marca del producto
  descripcion VARCHAR(200) NOT NULL,--descripcion del producto
  precio_venta DECIMAL(10,2) NOT NULL,--precio al que se vende el producto
  CONSTRAINT pk_producto PRIMARY KEY (id_producto),--llave primaria
  CONSTRAINT fk_producto_proveedor FOREIGN KEY (id_proveedor)
  REFERENCES PROVEEDOR(id_proveedor),--relacion con el proveedor
  CONSTRAINT uk_producto_codigo_barras UNIQUE (codigo_barras),-- código unico
  CONSTRAINT chk_producto_stock CHECK (stock >= 0),--check para el stock no sea negativo
  CONSTRAINT chk_producto_precios CHECK (precio_venta >= precio_compra),--para que el precio
  --sea mayor o igual a la compra
  CONSTRAINT chk_producto_tipo CHECK (tipo IN ('regalo', 'papeleria', 'impresion', 'recarga'))
  --check para los tipos validos
);
```

```
-- tabla VENTA
/*registra cada transaccion realizada
almacena ventas realizadas,enlaza con cliente empleados
dashbord proporciona la base de datos para calcular ingresos del mes y relacionar
ventas con empleados

*/

CREATE TABLE VENTA (
  folio VARCHAR(8),--folio unico formato vent 001
  fecha_venta DATE NOT NULL DEFAULT CURRENT_DATE,--fecha en la que se realiza la venta
  total DECIMAL(10,2) NOT NULL,--total de la venta
  id_empleado INT NOT NULL,--empleado responsable de la venta
  id_cliente INT NOT NULL,--cliente que realizo la venta
  CONSTRAINT pk_venta PRIMARY KEY (folio),--llave primaria el folio
  CONSTRAINT fk_venta_empleado FOREIGN KEY (id_empleado)
  REFERENCES EMPLEADO(id_empleado),--relacion con empleado
  CONSTRAINT fk_venta_cliente FOREIGN KEY (id_cliente)
  REFERENCES CLIENTE(id_cliente),--relacion con cliente
  CONSTRAINT chk_venta_folio CHECK (folio ~ '^VENT-\d{3}$'),
  --check para la validacion del formato del folio 001
  CONSTRAINT chk_venta_total CHECK (total >= 0)
  --check para la validacion del total no puede ser negativo
);
```

tabla DETALLE\_VENTA

glosa Los productos vendidos en cada transaccion  
 mite analizar qué productos y en que cantidades se vende cada transaccion  
 hbord para calcular el top 3 productos más vendidos

```
ATE TABLE DETALLE_VENTA (
  id_detalle INT DEFAULT nextval('seq_detalle_venta'),--id unico autoincremental
  folio_venta VARCHAR(8) NOT NULL,--folio de venta asociada
  id_producto INT NOT NULL,--producto vendido
  cantidad INT NOT NULL,--cantidad del producto
  precio_unitario DECIMAL(10,2) NOT NULL,--precio unitario del producto
  subtotal DECIMAL(10,2) NOT NULL,--subtotal (cantidad * producto unitario)
  CONSTRAINT pk_detalle_venta PRIMARY KEY (id_detalle),--llave primaria id detalle
  CONSTRAINT fk_detalle_venta_venta FOREIGN KEY (folio_venta)
  REFERENCES VENTA(folio),--relacion con venta fk
  CONSTRAINT fk_detalle_venta_producto FOREIGN KEY (id_producto)
  REFERENCES PRODUCTO(id_producto),--relacion con producto
  CONSTRAINT chk_detalle_venta_cantidad CHECK (cantidad > 0),--check para cantidad positiva
  CONSTRAINT chk_detalle_venta_precios CHECK (precio_unitario > 0 AND subtotal > 0)
  --check para precios sean positivos
```

```

-- tabla ALERTA_STOCK
/*
para monitorear productos con bajo stock y mandar la alerta :v
*/

CREATE TABLE ALERTA_STOCK (
    id_alerta INT DEFAULT nextval('seq_alerta_stock'),--id unico autocrimental
    id_producto INT NOT NULL,--producto con bajo stock
    stock_actual INT NOT NULL,--cantidad actual en inventario
    fecha_alerta TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,--fecha de generacion de la al
    CONSTRAINT pk_alerta_stock PRIMARY KEY (id_alerta),--llave primaria
    CONSTRAINT fk_alerta_stock_producto FOREIGN KEY (id_producto)
    REFERENCES PRODUCTO(id_producto)--relacion con el producto
);

```

ejecución :

```

CREATE SEQUENCE
CREATE SEQUENCE
CREATE SEQUENCE
CREATE SEQUENCE
CREATE SEQUENCE
CREATE SEQUENCE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE VIEW
INSERT 0 10
INSERT 0 10

```

Input for the program (Optional)

INSERT 0 7										
id_proveedor	razon_social	telefono	estado	codigo_postal	colonia	calle	numero			
1	Papelería Moderna SA	5551234567	Ciudad de México	03800	Del Valle	Insurgentes	234			
2	Office Plus MX	5559876543	Jalisco	44100	Centro	Juárez	567			
3	Distribuidora Escolar	5552345678	Nuevo León	64000	Monterrey Centro	Reforma	890			
4	Material Express	5558765432	Puebla	72000	Centro	Morelos	123			
5	Suministros Escolares SA	5553334444	Estado de México	54030	Tlalnepantla	Hidalgo	456			
6	Papelería Nacional	5557778888	Querétaro	76000	Centro	Constituyentes	789			
7	Distribuidora Paper Max	5559990000	Guanajuato	37000	León Centro	Madero	234			
8	Mayorista de Oficina	5552223333	Veracruz	91000	Centro	Independencia	567			
9	Proveedora Escolar Elite	5554445555	Yucatán	97000	Mérida Centro	Reforma	890			
10	Papelería del Sureste	5556667777	Chiapas	29000	Tuxtla Centro	Zaragoza	123			
(10 rows)										
id_cliente	rfc	nombre	ap_paterno	ap_materno	email	estado	codigo_postal	colonia	calle	r
1	MELM800101XXX	Maria	Estrada	López	maria.e@email.com	CDMX	03100	Narvarte	División del Norte	4
2	GOHJ910215YYY	Juan	González	Hernández	juan.g@email.com	Jalisco	44130	Moderna	Vallarta	7
3	RASP850630ZZZ	Pedro	Ramírez	Sánchez	pedro.r@email.com	Puebla	72150	La Paz	Reforma	9
4	TORL780405WWW	Laura	Torres	Ruiz	laura.t@email.com	CDMX	03300	Roma	Durango	1
5	USPS000105ABC	Esmeralda	Hernández	Sanchez	Esmeralda.H@gmail.com	CDMX	03300	Guadalupe	Tlalpam	2
id_empleado	clave	nombre	ap_paterno	ap_materno	fecha_nacimiento	fecha_ingreso				
1	EMP001	Carlos	Martínez	López	1990-05-15	2020-01-15				
2	EMP002	Ana	García	Pérez	1988-08-20	2019-03-10				
3	EMP003	Roberto	Sánchez	Flores	1995-02-28	2021-06-01				
4	EMP004	Diana	López	Torres	1992-11-10	2018-09-15				
5	EMP005	Miguel	Rodríguez	Vega	1993-07-25	2022-01-10				
6	EMP006	Patricia	Torres	Sánchez	1991-03-18	2021-11-15				
(6 rows)										
id_producto	id_proveedor	codigo_barras	precio_compra	foto_url	fecha_compra	stock	tipo	marca	descripcion	
1	1	7501234567890	15.50	cuaderno1.jpg	2024-01-15	50	papeleria	Scribe	Cuaderno profesional raya 10	
2	1	7501234567891	8.20	lapiz1.jpg	2024-01-15	100	papeleria	Bic	Lápiz #2	
3	2	7501234567892	45.00	regalo1.jpg	2024-01-20	20	regalo	Pelikan	Set de plumas fuente	
4	2	7501234567893	150.00	impresion1.jpg	2024-01-20	5	impresion	HP	Cartucho tinta negra	
5	1	7501234567894	12.50	libreta1.jpg	2024-01-15	40	papeleria	Norma	Libreta francesa rayada	
6	2	7501234567895	25.00	marcadores1.jpg	2024-01-15	30	papeleria	Crayola	Set marcadores 12 colores	
7	3	7501234567896	85.00	regalo2.jpg	2024-01-20	15	regalo	Mont Blanc	Porta plumas ejecutivo	
8	4	7501234567897	200.00	impresion2.jpg	2024-01-20	8	impresion	Epson	Tinta color magenta	

```

CREATE FUNCTION
  folio | fecha | cliente_nombre | empleado_nombre |
-----+-----+-----+-----+
VENT-001 | 2024-01-15 | María Estrada López | Carlos Martínez López | [{"producto" : "Cuaderno profesional raya 100h", "cantidad" : 2, "subtotal" : 50.00},
(1 row)

CREATE FUNCTION
  fecha | total_vendido | ganancia
-----+-----+-----
2024-01-15 | 110.00 | 38.00
2024-01-16 | 150.00 | 60.00
2024-01-17 | 60.00 | 19.00
(3 rows)

CREATE FUNCTION
CREATE TRIGGER
  id_producto | codigo_barras | descripcion | marca | stock | proveedor
-----+-----+-----+-----+-----+
(0 rows)

CREATE FUNCTION

```

codigo:

--Claves primarias (PK):

/\*

## 1. PROVEEDOR

- CONSTRAINT pk\_proveedor PRIMARY KEY (id\_proveedor)

## 2. CLIENTE

- CONSTRAINT pk\_cliente PRIMARY KEY (id\_cliente)

## 3. EMPLEADO

- CONSTRAINT pk\_empleado PRIMARY KEY (id\_empleado)

## 4. PRODUCTO

- CONSTRAINT pk\_producto PRIMARY KEY (id\_producto)

## 5. VENTA

- CONSTRAINT pk\_venta PRIMARY KEY (folio)

## 6. DETALLE\_VENTA

- CONSTRAINT pk\_detalle\_venta PRIMARY KEY (id\_detalle)

## 7. ALERTA\_STOCK

- CONSTRAINT pk\_alerta\_stock PRIMARY KEY (id\_alerta)

\*/

--Claves externas (FK):

/\*

## 1. PRODUCTO

- CONSTRAINT fk\_producto\_proveedor FOREIGN KEY (id\_proveedor) REFERENCES  
PROVEEDOR(id\_proveedor)

## 2. VENTA

- CONSTRAINT fk\_venta\_empleado FOREIGN KEY (id\_empleado) REFERENCES  
EMPLEADO(id\_empleado)

- CONSTRAINT fk\_venta\_cliente FOREIGN KEY (id\_cliente) REFERENCES  
CLIENTE(id\_cliente)

## 3. DETALLE\_VENTA



- CONSTRAINT fk\_detalle\_venta\_venta FOREIGN KEY (folio\_venta) REFERENCES  
VENTA(folio)

- CONSTRAINT fk\_detalle\_venta\_producto FOREIGN KEY (id\_producto) REFERENCES  
PRODUCTO(id\_producto)

#### 4. ALERTA\_STOCK

- CONSTRAINT fk\_alerta\_stock\_producto FOREIGN KEY (id\_producto) REFERENCES  
PRODUCTO(id\_producto)

\*/

--secuencias para los IDs auto-incrementales

CREATE SEQUENCE seq\_proveedor START WITH 1;

CREATE SEQUENCE seq\_cliente START WITH 1;

CREATE SEQUENCE seq\_empleado START WITH 1;

CREATE SEQUENCE seq\_producto START WITH 1;

CREATE SEQUENCE seq\_detalle\_venta START WITH 1;

CREATE SEQUENCE seq\_alerta\_stock START WITH 1;

-- tabla PROVEEDOR

/\*

Almacena informacion basica de los proveedores identificamos el origen de cada producto

\*/

CREATE TABLE PROVEEDOR (

id\_proveedor INT DEFAULT nextval('seq\_proveedor'),--id unico autoincremental

razon\_social VARCHAR(100) NOT NULL,--nombre proveedor

telefono VARCHAR(15) NOT NULL,--telefono de contacto

estado VARCHAR(50) NOT NULL,--estado del proveedor

codigo\_postal VARCHAR(5) NOT NULL,--codigo postal

colonia VARCHAR(50) NOT NULL,--colonia

calle VARCHAR(100) NOT NULL,--calle

numero VARCHAR(10) NOT NULL,--numero exterior

CONSTRAINT pk\_proveedor PRIMARY KEY (id\_proveedor),--llave primaria

CONSTRAINT uk\_proveedor\_razon\_social UNIQUE (razon\_social)--razon social unica

);

--tabla CLIENTE

/\*

Almacena informacion de los clientes

permite asociar ventas con clientes y calcular ingresos del cliente

\*/

CREATE TABLE CLIENTE (

id\_cliente INT DEFAULT nextval('seq\_cliente'),--id unico autoincremental

```
    rfc VARCHAR(13) NOT NULL,--rfc del cliente
    nombre VARCHAR(50) NOT NULL,--nombre del cliente
    ap_paterno VARCHAR(50) NOT NULL,--apellido paterno
    ap_materno VARCHAR(50),--apellido materno opcional
    email VARCHAR(100) NOT NULL,--correo del cliente
    estado VARCHAR(50) NOT NULL,--estado del cliente
    codigo_postal VARCHAR(5) NOT NULL,-- código postal
    colonia VARCHAR(50) NOT NULL,--colonia
    calle VARCHAR(100) NOT NULL,--calle
    numero VARCHAR(10) NOT NULL,--numero exterior
    CONSTRAINT pk_cliente PRIMARY KEY (id_cliente),--llave primaria
    CONSTRAINT uk_cliente_rfc UNIQUE (rfc),--rfc unico
    CONSTRAINT uk_cliente_email UNIQUE (email)--email unico
);
```

```
-- tabla EMPLEADO
```

```
/*
```

```
almacena informacion de los empleados que realizan ventas
```

```
asociar las ventas al empleado permitiendo calcular ordenes registradas por cada uno
```

```
despues relacion con dasnoard para empleados con más ordenes
```

```
*/
```

```
CREATE TABLE EMPLEADO (  
    id_empleado INT DEFAULT nextval('seq_empleado'),--id unico auto incremental  
    clave VARCHAR(10) NOT NULL,-- clave unica para identificar al empleado  
    nombre VARCHAR(50) NOT NULL,--nombre del empleado  
    ap_paterno VARCHAR(50) NOT NULL,--apellido  
    ap_materno VARCHAR(50),--apellido materno opcional  
    fecha_nacimiento DATE NOT NULL,--fecha de nacimiento del empleado  
    fecha_ingreso DATE NOT NULL,-- fecha que ingreso al empleo  
    CONSTRAINT pk_empleado PRIMARY KEY (id_empleado),--llave priamria  
    CONSTRAINT uk_empleado_clave UNIQUE (clave),--clave unica  
    CONSTRAINT chk_empleado_fecha_ingreso CHECK (fecha_ingreso >= fecha_nacimiento)  
    --check para validar que la edad no tenga incoherencias  
);
```

-- tabla PRODUCTO

/\*

contiene datos del inventario para calcular ventas, utilidad y edentificar el top 3

\*/

CREATE TABLE PRODUCTO (

id\_producto INT DEFAULT nextval('seq\_producto'),--id unico autocrimental

id\_proveedor INT NOT NULL,--relacion con la tabla proveedor

codigo\_barras VARCHAR(13) NOT NULL,--codigo de barras unico

precio\_compra DECIMAL(10,2) NOT NULL,--precio de compra del producto en decimal

foto\_url VARCHAR(255),-- foto del producto en varchar para no subir fotos deberia ser blob

fecha\_compra DATE NOT NULL,--fecha de adquisicion

stock INT NOT NULL,--cantidad disponible en el inventario

tipo VARCHAR(20) NOT NULL,--tipo (papeleria, recarga, impresion , regalo)

marca VARCHAR(50) NOT NULL,--marca del producto

descripcion VARCHAR(200) NOT NULL,--descripcion del producto

precio\_venta DECIMAL(10,2) NOT NULL,--precio al que se vende el producto

CONSTRAINT pk\_producto PRIMARY KEY (id\_producto),--llave primaria

CONSTRAINT fk\_producto\_proveedor FOREIGN KEY (id\_proveedor)

REFERENCES PROVEEDOR(id\_proveedor),--relacion con el proveedor

CONSTRAINT uk\_producto\_codigo\_barras UNIQUE (codigo\_barras),-- código unico

CONSTRAINT chk\_producto\_stock CHECK (stock >= 0),--check para el stock no sea negativo

CONSTRAINT chk\_producto\_precios CHECK (precio\_venta >= precio\_compra),--para que el precio de venta

--sea mayor o igual a la compra

```
CONSTRAINT chk_producto_tipo CHECK (tipo IN ('regalo', 'papeleria', 'impresion',  
'recarga'))
```

```
--check para los tipos validos
```

```
);
```

```
-- tabla VENTA
```

```
/*registra cada transaccion realizada
```

```
almacena ventas realizadas,enlaza con cliente empleados
```

```
dashbord proporciona la base de datos para calcular ingresos del mes y relacionar
```

```
ventas con empleados
```

```
*/
```

```
CREATE TABLE VENTA (
```

```
folio VARCHAR(8),--folio unico formato vent 001
```

```
fecha_venta DATE NOT NULL DEFAULT CURRENT_DATE,--fecha en la que se realiza la  
venta
```

```
total DECIMAL(10,2) NOT NULL,--total de la venta
```

```
id_empleado INT NOT NULL,--empleado responsable de la venta
```

```
id_cliente INT NOT NULL,--cliente que realizo la venta
```

```
CONSTRAINT pk_venta PRIMARY KEY (folio),--llave primaria el folio
CONSTRAINT fk_venta_empleado FOREIGN KEY (id_empleado)
REFERENCES EMPLEADO(id_empleado),--relacion con empleado
CONSTRAINT fk_venta_cliente FOREIGN KEY (id_cliente)
REFERENCES CLIENTE(id_cliente),--relacion con cliente
CONSTRAINT chk_venta_folio CHECK (folio ~ '^VENT-\d{3}$'),
--check para la validacion del formato del folio 001
CONSTRAINT chk_venta_total CHECK (total >= 0)
--check para la validacion del total no puede ser negativo
);
```

```
-- tabla DETALLE_VENTA
```

```
/*
```

desglosa los productos vendidos en cada transaccion

permite analizar qué productos y en que cantidades se vende cada transaccion

dashbord para calcular el top 3 productos más vendidos

\*/

```
CREATE TABLE DETALLE_VENTA (  
    id_detalle INT DEFAULT nextval('seq_detalle_venta'),--id unico autoincremental  
    folio_venta VARCHAR(8) NOT NULL,--folio de venta asociada  
    id_producto INT NOT NULL,--producto vendido  
    cantidad INT NOT NULL,--cantidad del producto  
    precio_unitario DECIMAL(10,2) NOT NULL,--precio unitario del producto  
    subtotal DECIMAL(10,2) NOT NULL,--subtotal (cantidad * producto unitario)  
    CONSTRAINT pk_detalle_venta PRIMARY KEY (id_detalle),--llave primaria id detalle  
    CONSTRAINT fk_detalle_venta_venta FOREIGN KEY (folio_venta)  
    REFERENCES VENTA(folio),--relacion con venta fk  
    CONSTRAINT fk_detalle_venta_producto FOREIGN KEY (id_producto)  
    REFERENCES PRODUCTO(id_producto),--relacion con producto  
    CONSTRAINT chk_detalle_venta_cantidad CHECK (cantidad > 0),--check para cantidad  
    positiva  
    CONSTRAINT chk_detalle_venta_precios CHECK (precio_unitario > 0 AND subtotal > 0)  
    --check para precios sean positivos  
);
```



```
-- tabla ALERTA_STOCK
```

```
/*
```

```
para monitorear productos con bajo stock y mandar la alerta :v
```

```
*/
```

```
CREATE TABLE ALERTA_STOCK (
```

```
    id_alerta INT DEFAULT nextval('seq_alerta_stock'),--id unico autocrimental
```

```
    id_producto INT NOT NULL,--producto con bajo stock
```

```
    stock_actual INT NOT NULL,--cantidad actual en inventario
```

```
    fecha_alerta TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,--fecha de  
generacion de la alerta
```

```
    CONSTRAINT pk_alerta_stock PRIMARY KEY (id_alerta),--llave primaria
```

```
    CONSTRAINT fk_alerta_stock_producto FOREIGN KEY (id_producto)
```

```
    REFERENCES PRODUCTO(id_producto)--relacion con el producto
```

```
);
```

--índices para optimizar búsquedas frecuentes

/\*optimizar las consultas más frecuentes

buscar productos por código

filtrar ventas

consultar productos con bajo stock

\*/

--búsqueda para código de barras

CREATE INDEX idx\_producto\_codigo\_barras ON PRODUCTO(codigo\_barras);

--consultar ventas por fecha

CREATE INDEX idx\_venta\_fecha ON VENTA(fecha\_venta);

--filtrar productos vendidos

CREATE INDEX idx\_detalle\_venta\_producto ON DETALLE\_VENTA(id\_producto);

--condicional para identificar productos con stock bajo

CREATE INDEX idx\_producto\_stock\_bajo ON PRODUCTO(stock) WHERE stock < 3;

--vista para productos con stock bajo

/\*

para ver el monitoreo de productos con bajo stock

agrupando producto y proveedor

\*/

CREATE VIEW v\_productos\_stock\_bajo AS

SELECT

p.id\_producto,

p.codigo\_barras,

p.descripcion,

p.marca,

p.stock,

pr.razon\_social as proveedor

FROM PRODUCTO p

JOIN PROVEEDOR pr ON p.id\_proveedor = pr.id\_proveedor

WHERE p.stock < 3;

--inserts

--inserseccion por encabezados es mejor asi si tenemos la informacion como en este caso

-- 1. Primero: PROVEEDOR tenemos 10

INSERT INTO PROVEEDOR (razon\_social, telefono, estado, codigo\_postal, colonia, calle, numero) VALUES

('Papelería Moderna SA', '5551234567', 'Ciudad de México', '03800', 'Del Valle', 'Insurgentes', '234'),

('Office Plus MX', '5559876543', 'Jalisco', '44100', 'Centro', 'Juárez', '567'),

('Distribuidora Escolar', '5552345678', 'Nuevo León', '64000', 'Monterrey Centro', 'Reforma', '890'),

('Material Express', '5558765432', 'Puebla', '72000', 'Centro', 'Morelos', '123'),

('Suministros Escolares SA', '5553334444', 'Estado de México', '54030', 'Tlalnepantla', 'Hidalgo', '456'),

('Papelería Nacional', '5557778888', 'Querétaro', '76000', 'Centro', 'Constituyentes', '789'),

('Distribuidora Paper Max', '5559990000', 'Guanajuato', '37000', 'León Centro', 'Madero', '234'),

('Mayorista de Oficina', '5552223333', 'Veracruz', '91000', 'Centro', 'Independencia', '567'),

('Proveedora Escolar Elite', '5554445555', 'Yucatán', '97000', 'Mérida Centro', 'Reforma', '890'),

('Papelería del Sureste', '5556667777', 'Chiapas', '29000', 'Tuxtla Centro', 'Zaragoza', '123');

-- 2. Segundo: CLIENTE tenemos 40

```
INSERT INTO CLIENTE (rfc, nombre, ap_paterno, ap_materno, email, estado, codigo_postal,
colonia, calle, numero) VALUES

('MELM800101XXX', 'María', 'Estrada', 'López', 'maria.e@email.com', 'CDMX', '03100',
'Narvarte', 'División del Norte', '45'),

('GOHJ910215YYY', 'Juan', 'González', 'Hernández', 'juan.g@email.com', 'Jalisco', '44130',
'Moderna', 'Vallarta', '78'),

('RASP850630ZZZ', 'Pedro', 'Ramírez', 'Sánchez', 'pedro.r@email.com', 'Puebla', '72150', 'La
Paz', 'Reforma', '90'),

('TORL780405WWW', 'Laura', 'Torres', 'Ruiz', 'laura.t@email.com', 'CDMX', '03300', 'Roma',
'Durango', '123'),

('HERF890215ABC', 'Fernando', 'Hernández', 'Ramírez', 'fer.her@email.com', 'CDMX', '03200',
'Condesa', 'Tamaulipas', '234'),

('MARL900330DEF', 'Lucía', 'Martínez', 'Rodríguez', 'lucia.m@email.com', 'Jalisco', '44200',
'Providencia', 'Americas', '567'),

('SASG910445GHI', 'Gabriel', 'Sánchez', 'Silva', 'gab.san@email.com', 'Nuevo León', '64100',
'San Pedro', 'Vasconcelos', '890'),

('LOPD920560JKL', 'Daniel', 'López', 'Pérez', 'daniel.l@email.com', 'Puebla', '72200', 'San
Manuel', 'Universidad', '123'),
```

('GARM930675MNO', 'María', 'García', 'Mendoza', 'maria.g@email.com', 'Querétaro', '76130', 'Jurica', 'Real', '456'),

('TORJ940790PQR', 'José', 'Torres', 'Ruiz', 'jose.t@email.com', 'Guanajuato', '37200', 'Jardines', 'Luna', '789'),

('RAMC950805STU', 'Carmen', 'Ramírez', 'Morales', 'carmen.r@email.com', 'Veracruz', '91200', 'Costa Verde', 'Mar', '012'),

('FLOE960920VWX', 'Eduardo', 'Flores', 'Ortiz', 'edu.flo@email.com', 'Yucatán', '97130', 'Montejo', 'Paseo', '345'),

('DIAA970135YZA', 'Ana', 'Díaz', 'Aguirre', 'ana.d@email.com', 'Chiapas', '29200', 'La Salle', 'Primera', '678'),

('VAZM980250BCD', 'Miguel', 'Vázquez', 'Zavala', 'miguel.v@email.com', 'CDMX', '03300', 'Nápoles', 'Indiana', '901'),

('CRUL990365EFG', 'Laura', 'Cruz', 'Luna', 'laura.c@email.com', 'Jalisco', '44300', 'Chapalita', 'Niño', '234'),

('ROMJ000480HIJ', 'Juan', 'Romero', 'Méndez', 'juan.r@email.com', 'Nuevo León', '64200', 'Contry', 'Garza', '567'),

('HERR010595KLM', 'Rosa', 'Hernández', 'Reyes', 'rosa.h@email.com', 'Puebla', '72300', 'Angelópolis', 'Kepler', '890'),

('MARA020610NOP', 'Alberto', 'Martínez', 'Ramos', 'alberto.m@email.com', 'Querétaro', '76230', 'El Marqués', 'Real', '123'),

('GARP030725QRS', 'Patricia', 'García', 'Rosales', 'pat.gar@email.com', 'Guanajuato', '37300', 'Del Valle', 'Principal', '456'),

('SANC040840TUV', 'Carlos', 'Sánchez', 'Núñez', 'carlos.s@email.com', 'Veracruz', '91300', 'Mocambo', 'Playa', '789'),

('LOPV050955WXY', 'Victoria', 'López', 'Paredes', 'vicky.l@email.com', 'Yucatán', '97230', 'Norte', 'Maya', '012'),

('TORH061070ZAB', 'Hugo', 'Torres', 'Rivas', 'hugo.t@email.com', 'Chiapas', '29300', 'Real', 'Segunda', '345'),

('RAMA071185CDE', 'Adriana', 'Ramírez', 'Mata', 'adri.r@email.com', 'CDMX', '03400', 'Del Valle', 'Eugenia', '678'),

('FLOJ081200FGH', 'Jorge', 'Flores', 'Ochoa', 'jorge.f@email.com', 'Jalisco', '44400', 'Providencia', 'Toronto', '901'),

('DIAM091315IJK', 'Mariana', 'Díaz', 'Arellano', 'mari.d@email.com', 'Nuevo León', '64300', 'Cumbres', 'Monte', '234'),

('VAZS101430LMN', 'Sandra', 'Vázquez', 'Zambrano', 'sandra.v@email.com', 'Puebla', '72400', 'La Paz', 'Victoria', '567'),

('CRUG111545OPQ', 'Gustavo', 'Cruz', 'León', 'gus.cruz@email.com', 'Querétaro', '76330', 'Centro', 'Hidalgo', '890'),

('ROMI121660RST', 'Isabel', 'Romero', 'Miranda', 'isa.rom@email.com', 'Guanajuato', '37400', 'San Miguel', 'Luna', '123'),

('HERA131775UVW', 'Arturo', 'Hernández', 'Rangel', 'arturo.h@email.com', 'Veracruz', '91400', 'Centro', 'Mar', '456'),

('MARF141890XYZ', 'Francisco', 'Martínez', 'Ríos', 'fran.m@email.com', 'Yucatán', '97330', 'García Ginerés', 'Colón', '789'),

('GARE151905ABC', 'Elena', 'García', 'Ruvalcaba', 'elena.g@email.com', 'Chiapas', '29400',  
'Patria', 'Tercera', '012'),  
( 'SANR162020DEF', 'Roberto', 'Sánchez', 'Nava', 'rob.san@email.com', 'CDMX', '03500', 'Roma  
Sur', 'Durango', '345'),  
( 'LOPL172135GHI', 'Leticia', 'López', 'Ponce', 'leti.l@email.com', 'Jalisco', '44500', 'Country',  
'Montreal', '678'),  
( 'TORM182250JKL', 'Manuel', 'Torres', 'Rocha', 'manu.t@email.com', 'Nuevo León', '64400',  
'San Jerónimo', 'Sierra', '901'),  
( 'RAMC192365MNO', 'Cecilia', 'Ramírez', 'Medina', 'ceci.r@email.com', 'Puebla', '72500',  
'Huexotitla', 'Universidad', '234'),  
( 'FLOA202480PQR', 'Antonio', 'Flores', 'Ojeda', 'antonio.f@email.com', 'Querétaro', '76430',  
'Álamos', 'Central', '567'),  
( 'DIAC212595STU', 'Claudia', 'Díaz', 'Ávila', 'clau.d@email.com', 'Guanajuato', '37500',  
'Jardines', 'Sol', '890'),  
( 'VAZJ222610VWX', 'Javier', 'Vázquez', 'Zúñiga', 'javi.v@email.com', 'Veracruz', '91500',  
'Reforma', 'Playa', '123'),  
( 'CRUV232725YZA', 'Verónica', 'Cruz', 'Lara', 'vero.c@email.com', 'Yucatán', '97430', 'México',  
'Progreso', '456'),  
( 'ROMR242840BCD', 'Raúl', 'Romero', 'Mena', 'raul.r@email.com', 'Chiapas', '29500', 'Linda  
Vista', 'Cuarta', '789');



```
INSERT INTO EMPLEADO (clave, nombre, ap_paterno, ap_materno, fecha_nacimiento,  
fecha_ingreso) VALUES
```

```
('EMP001', 'Carlos', 'Martínez', 'López', '1990-05-15', '2020-01-15'),
```

```
('EMP002', 'Ana', 'García', 'Pérez', '1988-08-20', '2019-03-10'),
```

```
('EMP003', 'Roberto', 'Sánchez', 'Flores', '1995-02-28', '2021-06-01'),
```

```
('EMP004', 'Diana', 'López', 'Torres', '1992-11-10', '2018-09-15'),
```

```
('EMP005', 'Miguel', 'Rodríguez', 'Vega', '1993-07-25', '2022-01-10'),
```

```
('EMP006', 'Patricia', 'Torres', 'Sánchez', '1991-03-18', '2021-11-15');
```

-- 4. Cuarto: PRODUCTO tenemos 23

```
INSERT INTO PRODUCTO (id_proveedor, codigo_barras, precio_compra, foto_url,  
fecha_compra, stock, tipo, marca, descripcion, precio_venta) VALUES
```

```
(1, '7501234567890', 15.50, 'cuaderno1.jpg', '2024-01-15', 50, 'papeleria', 'Scribe', 'Cuaderno  
profesional raya 100h', 25.00),
```

```
(1, '7501234567891', 8.20, 'lapiz1.jpg', '2024-01-15', 100, 'papeleria', 'Bic', 'Lápiz #2', 12.00),
```

```
(2, '7501234567892', 45.00, 'regalo1.jpg', '2024-01-20', 20, 'regalo', 'Pelikan', 'Set de plumas  
fuente', 75.00),
```

```
(2, '7501234567893', 150.00, 'impresion1.jpg', '2024-01-20', 5, 'impresion', 'HP', 'Cartucho tinta  
negra', 250.00),
```

```
(1, '7501234567894', 12.50, 'libreta1.jpg', '2024-01-15', 40, 'papeleria', 'Norma', 'Libreta francesa  
rayada', 22.00),
```

(2, '7501234567895', 25.00, 'marcadores1.jpg', '2024-01-15', 30, 'papeleria', 'Crayola', 'Set marcadores 12 colores', 45.00),

(3, '7501234567896', 85.00, 'regalo2.jpg', '2024-01-20', 15, 'regalo', 'Mont Blanc', 'Porta plumas ejecutivo', 150.00),

(4, '7501234567897', 200.00, 'impresion2.jpg', '2024-01-20', 8, 'impresion', 'Epson', 'Tinta color magenta', 350.00),

(5, '7501234567898', 5.00, 'goma1.jpg', '2024-01-25', 100, 'papeleria', 'Pelikan', 'Goma blanca mediana', 10.00),

(6, '7501234567899', 15.00, 'folder1.jpg', '2024-01-25', 80, 'papeleria', 'Acco', 'Folder tamaño carta', 25.00),

(7, '7501234567900', 120.00, 'regalo3.jpg', '2024-01-30', 12, 'regalo', 'Parker', 'Set escritorio ejecutivo', 200.00),

(8, '7501234567901', 180.00, 'impresion3.jpg', '2024-01-30', 6, 'impresion', 'Canon', 'Cartucho tinta color', 300.00),

(9, '7501234567902', 8.00, 'lapicero1.jpg', '2024-02-01', 150, 'papeleria', 'Paper Mate', 'Lapicero punto medio', 15.00),

(10, '7501234567903', 35.00, 'cuaderno2.jpg', '2024-02-01', 45, 'papeleria', 'Scribe', 'Cuaderno profesional cuadro', 55.00),

(1, '7501234567904', 95.00, 'regalo4.jpg', '2024-02-05', 10, 'regalo', 'Cross', 'Pluma fuente básica', 160.00),

(2, '7501234567905', 250.00, 'impresion4.jpg', '2024-02-05', 4, 'impresion', 'HP', 'Toner negro básico', 400.00),

(3, '7501234567906', 7.50, 'borrador1.jpg', '2024-02-10', 200, 'papeleria', 'Factis', 'Borrador para lápiz', 12.00),

(4, '7501234567907', 18.00, 'clips1.jpg', '2024-02-10', 120, 'papeleria', 'MAE', 'Caja clips núm. 2', 30.00),

(5, '7501234567908', 150.00, 'regalo5.jpg', '2024-02-15', 8, 'regalo', 'Waterman', 'Pluma roller gel', 250.00),

(6, '7501234567909', 220.00, 'impresion5.jpg', '2024-02-15', 5, 'impresion', 'Brother', 'Cartucho tinta negra XL', 380.00),

(7, '7501234567910', 6.00, 'lapiz2.jpg', '2024-02-20', 180, 'papeleria', 'Maped', 'Lápiz HB triangular', 10.00),

(8, '7501234567911', 42.00, 'colores1.jpg', '2024-02-20', 35, 'papeleria', 'Prismacolor', 'Colores 12 pzas', 70.00),

(9, '7501234567912', 180.00, 'regalo6.jpg', '2024-02-25', 6, 'regalo', 'Sheaffer', 'Set pluma y portaminas', 300.00);

-- 5. Quinto: VENTA tenemos 14

INSERT INTO VENTA (folio, fecha\_venta, total, id\_empleado, id\_cliente) VALUES

('VENT-001', '2024-01-15', 125.00, 1, 1),

('VENT-002', '2024-01-16', 250.00, 2, 2),

('VENT-003', '2024-01-17', 75.00, 1, 3),

('VENT-004', '2024-01-18', 500.00, 3, 4),  
('VENT-026', '2024-01-19', 1200.00, 1, 1),  
('VENT-027', '2024-01-20', 800.00, 2, 2),  
('VENT-028', '2024-01-21', 1500.00, 1, 2),  
('VENT-029', '2024-01-22', 950.00, 2, 1),  
('VENT-030', '2024-01-23', 400.00, 1, 2),  
('VENT-031', '2024-01-24', 2000.00, 2, 1),  
('VENT-032', '2024-01-25', 1250.00, 1, 1),  
('VENT-033', '2024-01-26', 700.00, 2, 2),  
('VENT-034', '2024-01-27', 1750.00, 1, 2),  
('VENT-035', '2024-01-28', 1100.00, 2, 1);

-- 6. Sexto: DETALLE\_VENTA tenemos 17

INSERT INTO DETALLE\_VENTA (folio\_venta, id\_producto, cantidad, precio\_unitario,  
subtotal) VALUES

('VENT-001', 1, 2, 25.00, 50.00),  
('VENT-001', 2, 5, 12.00, 60.00),  
('VENT-002', 3, 2, 75.00, 150.00),  
('VENT-003', 2, 5, 12.00, 60.00),  
('VENT-026', 1, 3, 120.00, 360.00),  
('VENT-026', 2, 4, 180.00, 720.00),  
('VENT-027', 3, 8, 100.00, 800.00),  
('VENT-028', 4, 5, 200.00, 1000.00),

```
('VENT-028', 5, 5, 100.00, 500.00),  
('VENT-029', 1, 5, 120.00, 600.00),  
('VENT-029', 2, 2, 175.00, 350.00),  
('VENT-030', 3, 4, 100.00, 400.00),  
('VENT-031', 6, 10, 200.00, 2000.00),  
('VENT-032', 1, 5, 250.00, 1250.00),  
('VENT-033', 7, 7, 100.00, 700.00),  
('VENT-034', 8, 5, 350.00, 1750.00),  
('VENT-035', 9, 5, 220.00, 1100.00);
```

-- 7. Séptimo: ALERTA\_STOCK

```
INSERT INTO ALERTA_STOCK (id_producto, stock_actual, fecha_alerta) VALUES
```

```
(1, 2, CURRENT_TIMESTAMP),
```

```
(3, 1, CURRENT_TIMESTAMP),
```

```
(4, 2, CURRENT_TIMESTAMP),
```

```
(5, 1, CURRENT_TIMESTAMP ), -- Producto 5 casi sin stock
```

```
(6, 3, CURRENT_TIMESTAMP), -- Producto 6 con bajo stock
```

```
(7, 0, CURRENT_TIMESTAMP), -- Producto 7 sin stock disponible
```

```
(8, 4, CURRENT_TIMESTAMP); -- Producto 8 con bajo stock
```

```
--select para ver las tablas
```

```
select * from PROVEEDOR;
```

```
select * from CLIENTE;
```

```
select * from EMPLEADO;
```

```
select * from PRODUCTO;
```

```
select * from VENTA;
```

```
select * from DETALLE_VENTA;
```

```
select * from ALERTA_STOCK;
```

```
-- Ingresos del mes
```

```
--calcular metricas financieras para el mes actual
```

```
WITH metricas_mes AS (  
    SELECT  
        SUM(p.precio_compra * dv.cantidad) as total_invertido,--costo total de productos vendidos  
        SUM(dv.subtotal) as total_ingresos,--total generado por las ventas  
        SUM(dv.subtotal - (p.precio_compra * dv.cantidad)) as ganancias--ganancias  
(ingresos-costos)  
    FROM VENTA v  
    JOIN DETALLE_VENTA dv ON v.folio = dv.folio_venta--union para obtener detalles de  
ventas  
    JOIN PRODUCTO p ON dv.id_producto = p.id_producto--relaciona productos con ventas  
    WHERE DATE_TRUNC('month', v.fecha_venta) = DATE_TRUNC('month',  
CURRENT_DATE)  
    --filtra para el mes actual  
)  
SELECT  
    total_invertido as "Inversión",  
    total_ingresos as "Ingresos",  
    ganancias as "Ganancias"  
FROM metricas_mes;
```

-- Top 3 artículos más vendidos

```
SELECT
    p.descripcion,
    p.marca,
    SUM(dv.cantidad) as total_vendido,
    SUM(dv.subtotal) as total_ingresos
FROM PRODUCTO p
JOIN DETALLE_VENTA dv ON p.id_producto = dv.id_producto
JOIN VENTA v ON dv.folio_venta = v.folio
WHERE DATE_TRUNC('month', v.fecha_venta) = DATE_TRUNC('month',
CURRENT_DATE)
GROUP BY p.id_producto, p.descripcion, p.marca
ORDER BY total_vendido DESC
LIMIT 3;
```

-- Empleados con más órdenes

--obtener empleados con más ventas durante el mes actual



```
SELECT

    e.nombre,--nombre del empleado

    e.ap_paterno,--apellido paterno

    COUNT(v.folio) as total_ordenes,--total de ventas registradas por el empleado

    SUM(v.total) as total_ventas--total generado por sus ventas

FROM EMPLEADO e

JOIN VENTA v ON e.id_empleado = v.id_empleado--relacionar ventas con empleados

WHERE DATE_TRUNC('month', v.fecha_venta) = DATE_TRUNC('month',

CURRENT_DATE)

--filtrar por mes actual

GROUP BY e.id_empleado, e.nombre, e.ap_paterno--agrupar por empleado

ORDER BY total_ordenes DESC;--ordenar por total de ordenes de mayor a menor
```

--funcion para generar factura

```
CREATE OR REPLACE FUNCTION generar_factura(folio_venta_param VARCHAR)

RETURNS TABLE(

    folio VARCHAR,

    fecha DATE,
```

```
cliente_nombre VARCHAR,

empleado_nombre VARCHAR,

productos JSON,

total DECIMAL

) AS $$

BEGIN

RETURN QUERY

SELECT

    v.folio::VARCHAR,

    v.fecha_venta,

    CONCAT(c.nombre, ' ', c.ap_paterno, ' ', COALESCE(c.ap_materno, ''))::VARCHAR AS

cliente_nombre,

    CONCAT(e.nombre, ' ', e.ap_paterno, ' ', COALESCE(e.ap_materno, ''))::VARCHAR AS

empleado_nombre,

    json_agg(json_build_object(

        'producto', p.descripcion,

        'cantidad', dv.cantidad,

        'subtotal', dv.subtotal))::JSON AS productos,

    v.total

FROM VENTA v

JOIN CLIENTE c ON v.id_cliente = c.id_cliente

JOIN EMPLEADO e ON v.id_empleado = e.id_empleado

JOIN DETALLE_VENTA dv ON v.folio = dv.folio_venta
```

```
JOIN PRODUCTO p ON dv.id_producto = p.id_producto

WHERE v.folio = folio_venta_param

GROUP BY v.folio, v.fecha_venta, cliente_nombre, empleado_nombre, v.total;

END;

$$ LANGUAGE plpgsql;
```

--para ver la factura

```
SELECT * FROM generar_factura('VENT-001');
```

--punto 2

```
CREATE OR REPLACE FUNCTION reporte_ganancias(fecha_inicio DATE, fecha_fin DATE)
RETURNS TABLE(
    fecha DATE,
    total_vendido DECIMAL,
    ganancia DECIMAL
) AS $$
BEGIN
    RETURN QUERY
```

```
SELECT
    v.fecha_venta,
    SUM(dv.subtotal) AS total_vendido,
    SUM(dv.subtotal - (p.precio_compra * dv.cantidad)) AS ganancia
FROM VENTA v
JOIN DETALLE_VENTA dv ON v.folio = dv.folio_venta
JOIN PRODUCTO p ON dv.id_producto = p.id_producto
WHERE v.fecha_venta BETWEEN fecha_inicio AND fecha_fin
GROUP BY v.fecha_venta;
END;
$$ LANGUAGE plpgsql;
```

--falta la fecha del 2024-01-18 lo hice asi para probar

```
SELECT * FROM reporte_ganancias('2024-01-15','2024-01-17');
```

--Actualización del stock al realizar una venta

-- Trigger para actualizar el stock después de cada venta

```
CREATE OR REPLACE FUNCTION actualizar_stock()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
-- Restar la cantidad vendida al stock

UPDATE PRODUCTO

SET stock = stock - NEW.cantidad

WHERE id_producto = NEW.id_producto;


-- Si el stock llega a cero, se aborta la transacción

IF (SELECT stock FROM PRODUCTO WHERE id_producto = NEW.id_producto) = 0

THEN

    RAISE EXCEPTION 'Stock agotado para el producto con id %', NEW.id_producto;

END IF;


-- Si el stock es menor a 3, se inserta una alerta

IF (SELECT stock FROM PRODUCTO WHERE id_producto = NEW.id_producto) < 3

THEN

    INSERT INTO ALERTA_STOCK (id_producto, stock_actual)

    VALUES (NEW.id_producto, (SELECT stock FROM PRODUCTO WHERE id_producto =

NEW.id_producto));

END IF;


RETURN NEW;

END;

$$ LANGUAGE plpgsql;
```

-- Asociamos el trigger con la tabla DETALLE\_VENTA

CREATE TRIGGER trg\_actualizar\_stock

AFTER INSERT ON DETALLE\_VENTA

FOR EACH ROW

EXECUTE FUNCTION actualizar\_stock();

--Reporte de productos con bajo stock

-- para asegurarnos de que siempre refleje la última información.

SELECT \* FROM v\_productos\_stock\_bajo;

--

CREATE OR REPLACE FUNCTION obtener\_utilidad(codigo\_barras\_param VARCHAR)

RETURNS TABLE(

    producto\_nombre VARCHAR,--para que salga el nombre del producto

    utilidad DECIMAL--para la utilidad

) AS \$\$

BEGIN

    RETURN QUERY

    SELECT

```
    descripcion AS producto_nombre, -- Descripción del producto

    (precio_venta - precio_compra) AS utilidad -- Cálculo de la utilidad

FROM PRODUCTO

WHERE codigo_barras = codigo_barras_param;

-- Si no se encuentra el producto, lanzar excepción

IF NOT FOUND THEN

    RAISE EXCEPTION 'Producto con código de barras % no encontrado',
codigo_barras_param;

END IF;

END;

$$ LANGUAGE plpgsql;

-- Ejemplo de uso

SELECT obtener_utilidad('7501234567890');

SELECT obtener_utilidad('7501234567892');
```

