

```
CREATE TABLE USUARIO(  
    nombre VARCHAR2(50) NOT NULL,  
    passhash1 NUMBER(10) NOT NULL,  
    passhash2 NUMBER(10) NOT NULL,  
    nivel NUMBER(2) NOT NULL,  
    CONSTRAINT pk_usuario PRIMARY KEY(nombre)  
);
```

```
CREATE TABLE CATEGORIA(  
    id VARCHAR2(40) NOT NULL,  
    descripcion VARCHAR2(100) NOT NULL,  
    CONSTRAINT pk_categoria PRIMARY KEY(id)  
);
```

```
CREATE TABLE PRODUCTO(  
    id NUMBER(10) NOT NULL,  
    nombre VARCHAR2(40) NOT NULL,  
    descripcion VARCHAR2(100) NOT NULL,  
    precio NUMBER(10,2) NOT NULL,  
    categoria_id VARCHAR2(40) NOT NULL,  
    inventario NUMBER(10) NOT NULL,  
    CONSTRAINT pk_producto PRIMARY KEY(id),  
    CONSTRAINT fk_categoria FOREIGN KEY(categoria_id) REFERENCES CATEGORIA(id)  
);
```

```
CREATE TABLE CLIENTE(  
    id NUMBER(10) NOT NULL,  
    nombre VARCHAR2(50) NOT NULL,  
    fecha_ultima_compra DATE,  
    correo VARCHAR2(50) NOT NULL,  
    direccion VARCHAR2(100) NOT NULL,  
    cantidad_compras NUMBER(10) NOT NULL,  
    CONSTRAINT pk_cliente PRIMARY KEY(id)
```

);

CREATE TABLE RECIBO(

id NUMBER(10) NOT NULL,

fecha DATE NOT NULL,

cliente_id NUMBER(10) NOT NULL,

CONSTRAINT pk_recibo PRIMARY KEY(id),

CONSTRAINT fk_cliente FOREIGN KEY(cliente_id) REFERENCES CLIENTE(id)

);

CREATE TABLE PRODUCTO_RECIBO(

producto_id NUMBER(10) NOT NULL,

recibo_id NUMBER(10) NOT NULL,

cantidad NUMBER(10) NOT NULL,

CONSTRAINT pk_producto_recibo PRIMARY KEY(producto_id, recibo_id),

CONSTRAINT fk_producto_recibo_producto FOREIGN KEY(producto_id) REFERENCES PRODUCTO(id),

CONSTRAINT fk_producto_recibo_recibo FOREIGN KEY(recibo_id) REFERENCES RECIBO(id)

);

CREATE TABLE RECIBO_DETALLE(

id NUMBER(10) NOT NULL,

recibo_id NUMBER(10) NOT NULL,

monto NUMBER(10,2) NOT NULL,

cantidad NUMBER(10) NOT NULL,

metodo_pago VARCHAR2(30) NOT NULL,

descripcion VARCHAR2(100) NOT NULL,

CONSTRAINT pk_recibo_detalle PRIMARY KEY(id),

CONSTRAINT fk_recibo_detalle_recibo FOREIGN KEY(recibo_id) REFERENCES RECIBO(id)

);

CREATE TABLE PROVEEDOR(

id NUMBER(10) NOT NULL,

nombre VARCHAR2(50) NOT NULL,

```
descripcion VARCHAR2(100) NOT NULL,  
direccion VARCHAR2(100) NOT NULL,  
CONSTRAINT pk_proveedor PRIMARY KEY(id)  
);
```

```
CREATE TABLE COMPRA_PRODUCTO(  
    id NUMBER(10) NOT NULL,  
    producto_id NUMBER(10) NOT NULL,  
    proveedor_id NUMBER(10) NOT NULL,  
    fecha DATE NOT NULL,  
    CONSTRAINT pk_compra_producto PRIMARY KEY(id),  
    CONSTRAINT fk_producto FOREIGN KEY(producto_id) REFERENCES PRODUCTO(id),  
    CONSTRAINT fk_proveedor FOREIGN KEY(proveedor_id) REFERENCES PROVEEDOR(id)  
);
```

```
CREATE TABLE COMPRA_DETALLE(  
    id NUMBER(10) NOT NULL,  
    compra_producto_id NUMBER(10) NOT NULL,  
    cantidad NUMBER(10) NOT NULL,  
    monto NUMBER(10,2) NOT NULL,  
    descripcion VARCHAR2(100) NOT NULL,  
    CONSTRAINT pk_compra_detalle PRIMARY KEY(id),  
    CONSTRAINT fk_compra_producto FOREIGN KEY(compra_producto_id) REFERENCES  
COMPRA_PRODUCTO(id)  
);
```

```
CREATE TABLE BITACORA_PRODUCTO(  
    id NUMBER(10) NOT NULL,  
    fecha TIMESTAMP NOT NULL,  
    descripcion VARCHAR2(600) NOT NULL,  
    usuario_db VARCHAR2(50) NOT NULL,  
    CONSTRAINT pk_bitacora_producto PRIMARY KEY(id)  
);
```

/

```
CREATE SEQUENCE seq_producto START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_cliente START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_recibo START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_recibo_detalle START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_proveedor START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_compra_producto START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_compra_detalle START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_bitacora_producto START WITH 1 INCREMENT BY 1;
```

/

```
CREATE OR REPLACE TRIGGER bitacora_producto_insert
AFTER INSERT ON PRODUCTO
FOR EACH ROW
  DECLARE
    producto_id NUMBER(10);
    producto_nombre VARCHAR2(40);
  BEGIN
    producto_id := :NEW.id;
    producto_nombre := :NEW.nombre;
    INSERT INTO BITACORA_PRODUCTO(id, fecha, descripcion, usuario_db)
      VALUES(seq_bitacora_producto.NEXTVAL, SYSTIMESTAMP,'Se insertó el producto ' || producto_nombre || '
con id ' || TO_CHAR(producto_id), USER);
  END;
```

/

```
CREATE OR REPLACE TRIGGER bitacora_producto_update
AFTER UPDATE ON PRODUCTO
FOR EACH ROW
  DECLARE
    producto_id NUMBER(10);
```

```

    producto_nombre VARCHAR2(40);
    mensaje VARCHAR2(600);
    producto_nombre_nuevo VARCHAR2(40);
    producto_nombre_anterior VARCHAR2(40);
    producto_descripcion_nuevo VARCHAR2(100);
    producto_descripcion_anterior VARCHAR2(100);
    producto_precio_nuevo NUMBER(10,2);
    producto_precio_anterior NUMBER(10,2);
    producto_categoria_nuevo VARCHAR2(10);
    producto_categoria_anterior VARCHAR2(10);
    producto_inventario_nuevo NUMBER(10);
    producto_inventario_anterior NUMBER(10);
BEGIN
    producto_id := :NEW.id;
    producto_nombre := :OLD.nombre;
    producto_nombre_nuevo := :NEW.nombre;
    producto_nombre_anterior := :OLD.nombre;
    producto_descripcion_nuevo := :NEW.descripcion;
    producto_descripcion_anterior := :OLD.descripcion;
    producto_precio_nuevo := :NEW.precio;
    producto_precio_anterior := :OLD.precio;
    producto_categoria_nuevo := :NEW.categoria_id;
    producto_categoria_anterior := :OLD.categoria_id;
    producto_inventario_nuevo := :NEW.inventario;
    producto_inventario_anterior := :OLD.inventario;

    mensaje := 'Se actualizó el producto ' || producto_nombre || ' con id ' || TO_CHAR(producto_id);

    IF producto_nombre_nuevo <> producto_nombre_anterior THEN
        mensaje := mensaje || ' '; su nombre cambió de '" || producto_nombre_anterior || '" a '" ||
        producto_nombre_nuevo || '"';
    END IF;

```

```

IF producto_descripcion_nuevo <> producto_descripcion_anterior THEN

    mensaje := mensaje || ' '; su descripción cambió de ' ' || producto_descripcion_anterior || ' ' a ' ' ||
producto_descripcion_nuevo || ' ';

END IF;

IF producto_precio_nuevo <> producto_precio_anterior THEN

    mensaje := mensaje || ' '; su precio cambió de ' ' || TO_CHAR(producto_precio_anterior) || ' ' a ' ' ||
TO_CHAR(producto_precio_nuevo);

END IF;

IF producto_categoria_nuevo <> producto_categoria_anterior THEN

    mensaje := mensaje || ' '; su categoría cambió de ' ' || producto_categoria_anterior || ' ' a ' ' ||
producto_categoria_nuevo;

END IF;

IF producto_inventario_nuevo <> producto_inventario_anterior THEN

    mensaje := mensaje || ' '; su inventario cambió de ' ' || TO_CHAR(producto_inventario_anterior) || ' ' a ' ' ||
TO_CHAR(producto_inventario_nuevo);

END IF;

INSERT INTO BITACORA_PRODUCTO(id, fecha, descripcion, usuario_db)
VALUES(seq_bitacora_producto.NEXTVAL, SYSTIMESTAMP, mensaje, USER);

END;

/

```

```

CREATE OR REPLACE TRIGGER bitacora_producto_delete
AFTER DELETE ON PRODUCTO
FOR EACH ROW

DECLARE

    producto_id NUMBER(10);

    producto_nombre VARCHAR2(40);

BEGIN

    producto_id := :OLD.id;

    producto_nombre := :OLD.nombre;

    INSERT INTO BITACORA_PRODUCTO(id, fecha, descripcion, usuario_db)

```

```

VALUES(seq_bitacora_producto.NEXTVAL, SYSTIMESTAMP, 'Se eliminó el producto ' || producto_nombre || '
con id ' || TO_CHAR(producto_id), USER);

END;

/

```

```

CREATE OR REPLACE TRIGGER recibo_insert

AFTER INSERT ON RECIBO

FOR EACH ROW

DECLARE

    recibo_fecha DATE;

    recibo_cliente_id NUMBER(10);

BEGIN

    recibo_fecha := :NEW.fecha;

    recibo_cliente_id := :NEW.cliente_id;

    UPDATE CLIENTE

    SET fecha_ultima_compra = recibo_fecha, cantidad_compras = cantidad_compras + 1

    WHERE id = recibo_cliente_id;

END;

/

```

```

CREATE OR REPLACE PACKAGE paquete_modificar AS

    PROCEDURE insertar_producto(p_nombre VARCHAR2, p_descripcion VARCHAR2, p_precio NUMBER,
p_categoria_id VARCHAR2, p_inventario NUMBER);

    PROCEDURE actualizar_producto(p_id NUMBER, p_nombre VARCHAR2, p_descripcion VARCHAR2, p_precio
NUMBER, p_categoria_id VARCHAR2, p_inventario NUMBER);

    PROCEDURE eliminar_producto(p_id NUMBER);

    PROCEDURE insertar_cliente(p_nombre VARCHAR2, p_fecha_ultima_compra DATE, p_correo VARCHAR2,
p_direccion VARCHAR2, p_cantidad_compras NUMBER);

    PROCEDURE actualizar_cliente(p_id NUMBER, p_nombre VARCHAR2, p_fecha_ultima_compra DATE,
p_correo VARCHAR2, p_direccion VARCHAR2, p_cantidad_compras NUMBER);

    PROCEDURE eliminar_cliente(p_id NUMBER);

    PROCEDURE insertar_recibo(p_fecha DATE, p_cliente_id NUMBER);

    PROCEDURE actualizar_recibo(p_id NUMBER, p_fecha DATE, p_cliente_id NUMBER);

    PROCEDURE eliminar_recibo(p_id NUMBER);

END;

```

```

PROCEDURE insertar_proveedor(p_nombre VARCHAR2, p_descripcion VARCHAR2, p_direccion VARCHAR2);

PROCEDURE actualizar_proveedor(p_id NUMBER, p_nombre VARCHAR2, p_descripcion VARCHAR2,
p_direccion VARCHAR2);

PROCEDURE eliminar_proveedor(p_id NUMBER);

PROCEDURE insertar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER, p_cantidad
NUMBER);

PROCEDURE actualizar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER, p_cantidad
NUMBER);

PROCEDURE eliminar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER);

PROCEDURE eliminar_producto_recibo_completo(p_recibo_id NUMBER);

PROCEDURE insertar_recibo_detalle(p_recibo_id NUMBER, p_monto NUMBER, p_cantidad NUMBER,
p_metodo_pago VARCHAR2, p_descripcion VARCHAR2);

PROCEDURE actualizar_recibo_detalle(p_id NUMBER, p_recibo_id NUMBER, p_monto NUMBER, p_cantidad
NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2);

PROCEDURE eliminar_recibo_detalle(p_id NUMBER);

PROCEDURE insertar_recibo_completo(p_fecha DATE, p_cliente_id NUMBER, p_monto NUMBER, p_cantidad
NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2);

PROCEDURE insertar_compra_producto(p_producto_id NUMBER, p_proveedor_id NUMBER, p_fecha DATE);

PROCEDURE actualizar_compra_producto(p_id NUMBER, p_producto_id NUMBER, p_proveedor_id
NUMBER, p_fecha DATE);

PROCEDURE eliminar_compra_producto(p_id NUMBER);

PROCEDURE insertar_compra_detalle(p_compra_producto_id NUMBER, p_cantidad NUMBER, p_monto
NUMBER, p_descripcion VARCHAR2);

PROCEDURE actualizar_compra_detalle(p_id NUMBER, p_compra_producto_id NUMBER, p_cantidad
NUMBER, p_monto NUMBER, p_descripcion VARCHAR2);

PROCEDURE eliminar_compra_detalle(p_id NUMBER);

PROCEDURE insertar_compra_producto_completo(p_producto_id NUMBER, p_proveedor_id NUMBER,
p_fecha DATE, p_cantidad NUMBER, p_monto NUMBER, p_descripcion VARCHAR2);

PROCEDURE actualizar_recibo_completo(p_id NUMBER, p_fecha DATE, p_cliente_id NUMBER, p_monto
NUMBER, p_cantidad NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2);

PROCEDURE eliminar_recibo_completo(p_id NUMBER);

PROCEDURE actualizar_compra_producto_completo(p_id NUMBER, p_producto_id NUMBER,
p_proveedor_id NUMBER, p_fecha DATE, p_cantidad NUMBER, p_monto NUMBER, p_descripcion VARCHAR2);

PROCEDURE eliminar_compra_producto_completo(p_id NUMBER);

END;

/

```



```

CREATE OR REPLACE PACKAGE paquete_select AS

    FUNCTION obtener_productos RETURN SYS_REFCURSOR;

    FUNCTION obtener_producto(p_id NUMBER) RETURN SYS_REFCURSOR;

    FUNCTION obtener_productos_categoria RETURN SYS_REFCURSOR; -- incluir join con categoria

    FUNCTION obtener_producto_categoria(p_id NUMBER) RETURN SYS_REFCURSOR; -- incluir join con
categoria ACA

    FUNCTION obtener_bitacora_productos RETURN SYS_REFCURSOR;

    FUNCTION obtener_bitacora_producto(p_id NUMBER) RETURN SYS_REFCURSOR;

    FUNCTION obtener_clientes RETURN SYS_REFCURSOR;

    FUNCTION obtener_cliente(p_id NUMBER) RETURN SYS_REFCURSOR;

    FUNCTION obtener_recibos RETURN SYS_REFCURSOR;

    FUNCTION obtener_recibo(p_id NUMBER) RETURN SYS_REFCURSOR; -- incluir join con recibo_detalle

    FUNCTION obtener_proveedores RETURN SYS_REFCURSOR;

    FUNCTION obtener_proveedor(p_id NUMBER) RETURN SYS_REFCURSOR;

    FUNCTION obtener_compra_productos RETURN SYS_REFCURSOR;

    FUNCTION obtener_compra_producto(p_id NUMBER) RETURN SYS_REFCURSOR; -- incluir join con
compra_detalle

    FUNCTION obtener_articulos_vendido_mes(anio_mes DATE) RETURN NUMBER; -- retorna la cantidad
vendida por mes en total

    FUNCTION obtener_articulo_vendido_mes(p_id NUMBER, anio_mes DATE) RETURN NUMBER;

    FUNCTION obtener_producto_recibo_todos RETURN SYS_REFCURSOR;

    FUNCTION obtener_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER) RETURN
SYS_REFCURSOR;

    FUNCTION obtener_productos_recibo(p_recibo_id NUMBER) RETURN SYS_REFCURSOR;

    FUNCTION obtener_producto_recibos(p_producto_id NUMBER) RETURN SYS_REFCURSOR;

END;

/

```

```

CREATE OR REPLACE PACKAGE BODY paquete_modificar AS

    PROCEDURE insertar_producto(p_nombre VARCHAR2, p_descripcion VARCHAR2, p_precio NUMBER,
p_categoria_id VARCHAR2, p_inventario NUMBER) AS

    BEGIN

        INSERT INTO PRODUCTO(id, nombre, descripcion, precio, categoria_id, inventario)

        VALUES(seq_producto.NEXTVAL, p_nombre, p_descripcion, p_precio, p_categoria_id, p_inventario);

    EXCEPTION

```

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20001, 'Error al insertar producto');

END insertar_producto;

PROCEDURE actualizar_producto(p_id NUMBER, p_nombre VARCHAR2, p_descripcion VARCHAR2, p_precio
NUMBER, p_categoria_id VARCHAR2, p_inventario NUMBER) AS

BEGIN

UPDATE PRODUCTO

SET nombre = p_nombre, descripcion = p_descripcion, precio = p_precio, categoria_id = p_categoria_id,
inventario = p_inventario

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20002, 'Error al actualizar producto');

END actualizar_producto;

PROCEDURE eliminar_producto(p_id NUMBER) AS

BEGIN

DELETE FROM PRODUCTO

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20003, 'Error al eliminar producto');

END eliminar_producto;

PROCEDURE insertar_cliente(p_nombre VARCHAR2, p_fecha_ultima_compra DATE, p_correo VARCHAR2,
p_direccion VARCHAR2, p_cantidad_compras NUMBER) AS

BEGIN

INSERT INTO CLIENTE(id, nombre, fecha_ultima_compra, correo, direccion, cantidad_compras)

VALUES(seq_cliente.NEXTVAL, p_nombre, p_fecha_ultima_compra, p_correo, p_direccion,
p_cantidad_compras);

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20004, 'Error al insertar cliente');

END insertar_cliente;

PROCEDURE actualizar_cliente(p_id NUMBER, p_nombre VARCHAR2, p_fecha_ultima_compra DATE, p_correo VARCHAR2, p_direccion VARCHAR2, p_cantidad_compras NUMBER) AS

BEGIN

UPDATE CLIENTE

SET nombre = p_nombre, fecha_ultima_compra = p_fecha_ultima_compra, correo = p_correo, direccion = p_direccion, cantidad_compras = p_cantidad_compras

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20005, 'Error al actualizar cliente');

END actualizar_cliente;

PROCEDURE eliminar_cliente(p_id NUMBER) AS

BEGIN

DELETE FROM CLIENTE

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20006, 'Error al eliminar cliente');

END eliminar_cliente;

PROCEDURE insertar_recibo(p_fecha DATE, p_cliente_id NUMBER) AS

BEGIN

INSERT INTO RECIBO(id, fecha, cliente_id)

VALUES(seq_recibo.NEXTVAL, p_fecha, p_cliente_id);

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20007, 'Error al insertar recibo');

END insertar_recibo;

PROCEDURE actualizar_recibo(p_id NUMBER, p_fecha DATE, p_cliente_id NUMBER) AS

BEGIN

```

UPDATE RECIBO

SET fecha = p_fecha, cliente_id = p_cliente_id

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20008, 'Error al actualizar recibo');

END actualizar_recibo;

```

```

PROCEDURE eliminar_recibo(p_id NUMBER) AS

BEGIN

    DELETE FROM RECIBO

    WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20009, 'Error al eliminar recibo');

END eliminar_recibo;

```

```

PROCEDURE insertar_proveedor(p_nombre VARCHAR2, p_descripcion VARCHAR2, p_direccion VARCHAR2)
AS

BEGIN

    INSERT INTO PROVEEDOR(id, nombre, descripcion, direccion)

    VALUES(seq_proveedor.NEXTVAL, p_nombre, p_descripcion, p_direccion);

EXCEPTION

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20010, 'Error al insertar proveedor');

END insertar_proveedor;

```

```

PROCEDURE actualizar_proveedor(p_id NUMBER, p_nombre VARCHAR2, p_descripcion VARCHAR2,
p_direccion VARCHAR2) AS

BEGIN

    UPDATE PROVEEDOR

    SET nombre = p_nombre, descripcion = p_descripcion, direccion = p_direccion

    WHERE id = p_id;

EXCEPTION

```

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20011, 'Error al actualizar proveedor');

END actualizar_proveedor;

PROCEDURE eliminar_proveedor(p_id NUMBER) AS

BEGIN

DELETE FROM PROVEEDOR

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20012, 'Error al eliminar proveedor');

END eliminar_proveedor;

PROCEDURE insertar_recibo_detalle(p_recibo_id NUMBER, p_monto NUMBER, p_cantidad NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2) AS

BEGIN

INSERT INTO RECIBO_DETALLE(id, recibo_id, monto, cantidad, metodo_pago, descripcion)

VALUES(seq_recibo_detalle.NEXTVAL, p_recibo_id, p_monto, p_cantidad, p_metodo_pago, p_descripcion);

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20013, 'Error al insertar recibo detalle');

END insertar_recibo_detalle;

PROCEDURE actualizar_recibo_detalle(p_id NUMBER, p_recibo_id NUMBER, p_monto NUMBER, p_cantidad NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2) AS

BEGIN

UPDATE RECIBO_DETALLE

SET recibo_id = p_recibo_id, monto = p_monto, cantidad = p_cantidad, metodo_pago = p_metodo_pago, descripcion = p_descripcion

WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20014, 'Error al actualizar recibo detalle');

```
END actualizar_recibo_detalle;
```

```
PROCEDURE eliminar_recibo_detalle(p_id NUMBER) AS
```

```
BEGIN
```

```
    DELETE FROM RECIBO_DETALLE
```

```
    WHERE id = p_id;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20015, 'Error al eliminar recibo detalle');
```

```
END eliminar_recibo_detalle;
```

```
PROCEDURE insertar_recibo_completo(p_fecha DATE, p_cliente_id NUMBER, p_monto NUMBER, p_cantidad  
NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2) AS
```

```
BEGIN
```

```
    INSERT INTO RECIBO(id, fecha, cliente_id)
```

```
    VALUES(seq_recibo.NEXTVAL, p_fecha, p_cliente_id);
```

```
    INSERT INTO RECIBO_DETALLE(id, recibo_id, monto, cantidad, metodo_pago, descripcion)
```

```
    VALUES(seq_recibo_detalle.NEXTVAL, seq_recibo.CURRVAL, p_monto, p_cantidad, p_metodo_pago,  
p_descripcion);
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20016, 'Error al insertar recibo completo');
```

```
END insertar_recibo_completo;
```

```
PROCEDURE insertar_compra_producto(p_producto_id NUMBER, p_proveedor_id NUMBER, p_fecha DATE)  
AS
```

```
BEGIN
```

```
    INSERT INTO COMPRA_PRODUCTO(id, producto_id, proveedor_id, fecha)
```

```
    VALUES(seq_compra_producto.NEXTVAL, p_producto_id, p_proveedor_id, p_fecha);
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20017, 'Error al insertar compra producto');
```

```
END insertar_compra_producto;
```

```
PROCEDURE actualizar_compra_producto(p_id NUMBER, p_producto_id NUMBER, p_proveedor_id  
NUMBER, p_fecha DATE) AS
```

```
BEGIN
```

```
    UPDATE COMPRA_PRODUCTO
```

```
    SET producto_id = p_producto_id, proveedor_id = p_proveedor_id, fecha = p_fecha
```

```
    WHERE id = p_id;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20018, 'Error al actualizar compra producto');
```

```
END actualizar_compra_producto;
```

```
PROCEDURE eliminar_compra_producto(p_id NUMBER) AS
```

```
BEGIN
```

```
    DELETE FROM COMPRA_PRODUCTO
```

```
    WHERE id = p_id;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20019, 'Error al eliminar compra producto');
```

```
END eliminar_compra_producto;
```

```
PROCEDURE insertar_compra_detalle(p_compra_producto_id NUMBER, p_cantidad NUMBER, p_monto  
NUMBER, p_descripcion VARCHAR2) AS
```

```
BEGIN
```

```
    INSERT INTO COMPRA_DETALLE(id, compra_producto_id, cantidad, monto, descripcion)
```

```
    VALUES(seq_compra_detalle.NEXTVAL, p_compra_producto_id, p_cantidad, p_monto, p_descripcion);
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20020, 'Error al insertar compra detalle');
```

```
END insertar_compra_detalle;
```

```
PROCEDURE actualizar_compra_detalle(p_id NUMBER, p_compra_producto_id NUMBER, p_cantidad  
NUMBER, p_monto NUMBER, p_descripcion VARCHAR2) AS
```

```
BEGIN
```

```
    UPDATE COMPRA_DETALLE
```

```
        SET compra_producto_id = p_compra_producto_id, cantidad = p_cantidad, monto = p_monto, descripcion  
= p_descripcion
```

```
        WHERE id = p_id;
```

```
    EXCEPTION
```

```
        WHEN OTHERS THEN
```

```
            RAISE_APPLICATION_ERROR(-20021, 'Error al actualizar compra detalle');
```

```
END actualizar_compra_detalle;
```

```
PROCEDURE eliminar_compra_detalle(p_id NUMBER) AS
```

```
BEGIN
```

```
    DELETE FROM COMPRA_DETALLE
```

```
    WHERE id = p_id;
```

```
    EXCEPTION
```

```
        WHEN OTHERS THEN
```

```
            RAISE_APPLICATION_ERROR(-20022, 'Error al eliminar compra detalle');
```

```
END eliminar_compra_detalle;
```

```
PROCEDURE insertar_compra_producto_completo(p_producto_id NUMBER, p_proveedor_id NUMBER,  
p_fecha DATE, p_cantidad NUMBER, p_monto NUMBER, p_descripcion VARCHAR2) AS
```

```
BEGIN
```

```
    INSERT INTO COMPRA_PRODUCTO(id, producto_id, proveedor_id, fecha)
```

```
    VALUES(seq_compra_producto.NEXTVAL, p_producto_id, p_proveedor_id, p_fecha);
```

```
    INSERT INTO COMPRA_DETALLE(id, compra_producto_id, cantidad, monto, descripcion)
```

```
    VALUES(seq_compra_detalle.NEXTVAL, seq_compra_producto.CURRVAL, p_cantidad, p_monto,  
p_descripcion);
```

```
    EXCEPTION
```

```
        WHEN OTHERS THEN
```

```
            RAISE_APPLICATION_ERROR(-20023, 'Error al insertar compra producto completo');
```

```
END insertar_compra_producto_completo;
```

```
PROCEDURE insertar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER, p_cantidad  
NUMBER) AS
```

```
BEGIN
```

```
    INSERT INTO PRODUCTO_RECIBO(producto_id, recibo_id, cantidad)
```



```

VALUES(p_producto_id, p_recibo_id, p_cantidad);

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20024, 'Error al insertar producto recibo');

END insertar_producto_recibo;


PROCEDURE actualizar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER, p_cantidad
NUMBER) AS

BEGIN

    UPDATE PRODUCTO_RECIBO

    SET cantidad = p_cantidad

    WHERE producto_id = p_producto_id AND recibo_id = p_recibo_id;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20025, 'Error al actualizar producto recibo');

END actualizar_producto_recibo;


PROCEDURE eliminar_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER) AS

BEGIN

    DELETE FROM PRODUCTO_RECIBO

    WHERE producto_id = p_producto_id AND recibo_id = p_recibo_id;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20026, 'Error al eliminar producto recibo');

END eliminar_producto_recibo;


PROCEDURE eliminar_producto_recibo_completo(p_recibo_id NUMBER) AS

BEGIN

    DELETE FROM PRODUCTO_RECIBO

    WHERE recibo_id = p_recibo_id;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20027, 'Error al eliminar producto recibo completo');

```

```
END eliminar_producto_recibo_completo;
```

```
PROCEDURE actualizar_recibo_completo(p_id NUMBER, p_fecha DATE, p_cliente_id NUMBER, p_monto  
NUMBER, p_cantidad NUMBER, p_metodo_pago VARCHAR2, p_descripcion VARCHAR2) AS
```

```
BEGIN
```

```
    UPDATE RECIBO
```

```
    SET fecha = p_fecha, cliente_id = p_cliente_id
```

```
    WHERE id = p_id;
```

```
    UPDATE RECIBO_DETALLE
```

```
    SET monto = p_monto, cantidad = p_cantidad, metodo_pago = p_metodo_pago, descripcion =  
p_descripcion
```

```
    WHERE recibo_id = p_id;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20028, 'Error al actualizar recibo completo');
```

```
END actualizar_recibo_completo;
```

```
PROCEDURE eliminar_recibo_completo(p_id NUMBER) AS
```

```
BEGIN
```

```
    DELETE FROM RECIBO_DETALLE
```

```
    WHERE recibo_id = p_id;
```

```
    DELETE FROM RECIBO
```

```
    WHERE id = p_id;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20029, 'Error al eliminar recibo completo');
```

```
END eliminar_recibo_completo;
```

```
PROCEDURE actualizar_compra_producto_completo(p_id NUMBER, p_producto_id NUMBER,  
p_proveedor_id NUMBER, p_fecha DATE, p_cantidad NUMBER, p_monto NUMBER, p_descripcion VARCHAR2)  
AS
```

```
BEGIN
```

```
    UPDATE COMPRA_PRODUCTO
```

```
    SET producto_id = p_producto_id, proveedor_id = p_proveedor_id, fecha = p_fecha
```

```

WHERE id = p_id;

UPDATE COMPRA_DETALLE

SET cantidad = p_cantidad, monto = p_monto, descripcion = p_descripcion

WHERE compra_producto_id = p_id;

EXCEPTION

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20030, 'Error al actualizar compra producto completo');

END actualizar_compra_producto_completo;


PROCEDURE eliminar_compra_producto_completo(p_id NUMBER) AS

BEGIN

    DELETE FROM COMPRA_DETALLE

    WHERE compra_producto_id = p_id;

    DELETE FROM COMPRA_PRODUCTO

    WHERE id = p_id;

EXCEPTION

WHEN OTHERS THEN

    RAISE_APPLICATION_ERROR(-20031, 'Error al eliminar compra producto completo');

END eliminar_compra_producto_completo;


END paquete_modificar;

/

```

```

CREATE OR REPLACE PACKAGE BODY paquete_select AS

FUNCTION obtener_productos RETURN SYS_REFCURSOR AS

    productos SYS_REFCURSOR;

BEGIN

    OPEN productos FOR

    SELECT * FROM PRODUCTO;

    RETURN productos;

EXCEPTION

```

```
        WHEN OTHERS THEN

            RAISE_APPLICATION_ERROR(-20024, 'Error al obtener productos');

END obtener_productos;
```

```
FUNCTION obtener_producto(p_id NUMBER) RETURN SYS_REFCURSOR AS

    producto SYS_REFCURSOR;

BEGIN

    OPEN producto FOR

        SELECT * FROM PRODUCTO

        WHERE id = p_id;

    RETURN producto;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20025, 'Error al obtener producto');

END obtener_producto;
```

```
FUNCTION obtener_productos_categoria RETURN SYS_REFCURSOR AS

    productos_categoria SYS_REFCURSOR;

BEGIN

    OPEN productos_categoria FOR

        SELECT * FROM PRODUCTO

        INNER JOIN CATEGORIA

        ON PRODUCTO.categoria_id = CATEGORIA.id;

    RETURN productos_categoria;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20026, 'Error al obtener productos categoria');

END obtener_productos_categoria;
```

```
FUNCTION obtener_producto_categoria(p_id NUMBER) RETURN SYS_REFCURSOR AS --holssssssss

    producto_categoria SYS_REFCURSOR;

BEGIN

    OPEN producto_categoria FOR
```

```

SELECT * FROM PRODUCTO
INNER JOIN CATEGORIA
ON PRODUCTO.categoria_id = CATEGORIA.id
WHERE PRODUCTO.id = p_id;
RETURN producto_categoria;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20027, 'Error al obtener producto categoria');
END obtener_producto_categoria;

```

```

FUNCTION obtener_bitacora_productos RETURN SYS_REFCURSOR AS
    bitacora_productos SYS_REFCURSOR;
BEGIN
    OPEN bitacora_productos FOR
    SELECT * FROM BITACORA_PRODUCTO;
    RETURN bitacora_productos;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20028, 'Error al obtener bitacora productos');
END obtener_bitacora_productos;

```

```

FUNCTION obtener_bitacora_producto(p_id NUMBER) RETURN SYS_REFCURSOR AS
    bitacora_producto SYS_REFCURSOR;
BEGIN
    OPEN bitacora_producto FOR
    SELECT * FROM BITACORA_PRODUCTO
    WHERE id = p_id;
    RETURN bitacora_producto;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20029, 'Error al obtener bitacora producto');
END obtener_bitacora_producto;

```

```

FUNCTION obtener_clientes RETURN SYS_REFCURSOR AS
    clientes SYS_REFCURSOR;
BEGIN
    OPEN clientes FOR
    SELECT * FROM CLIENTE;
    RETURN clientes;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20030, 'Error al obtener clientes');
END obtener_clientes;

FUNCTION obtener_cliente(p_id NUMBER) RETURN SYS_REFCURSOR AS
    cliente SYS_REFCURSOR;
BEGIN
    OPEN cliente FOR
    SELECT * FROM CLIENTE
    WHERE id = p_id;
    RETURN cliente;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20031, 'Error al obtener cliente');
END obtener_cliente;

FUNCTION obtener_recibos RETURN SYS_REFCURSOR AS --6
    recibos SYS_REFCURSOR;
BEGIN
    OPEN recibos FOR
    SELECT * FROM RECIBO
    INNER JOIN RECIBO_DETALLE
    ON RECIBO.id = RECIBO_DETALLE.recibo_id;
    RETURN recibos;
EXCEPTION
    WHEN OTHERS THEN

```

```

        RAISE_APPLICATION_ERROR(-20032, 'Error al obtener recibos');
END obtener_recibos;

FUNCTION obtener_recibo(p_id NUMBER) RETURN SYS_REFCURSOR AS --7
    recibo SYS_REFCURSOR;
BEGIN
    OPEN recibo FOR
    SELECT * FROM RECIBO
    INNER JOIN RECIBO_DETALLE
    ON RECIBO.id = RECIBO_DETALLE.recibo_id
    WHERE RECIBO.id = p_id;
    RETURN recibo;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20033, 'Error al obtener recibo');
END obtener_recibo;

```

```

FUNCTION obtener_proveedores RETURN SYS_REFCURSOR AS --8
    proveedores SYS_REFCURSOR;
BEGIN
    OPEN proveedores FOR
    SELECT * FROM PROVEEDOR;
    RETURN proveedores;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20034, 'Error al obtener proveedores');
END obtener_proveedores;

```

```

FUNCTION obtener_proveedor(p_id NUMBER) RETURN SYS_REFCURSOR AS --9
    proveedor SYS_REFCURSOR;
BEGIN
    OPEN proveedor FOR
    SELECT * FROM PROVEEDOR

```

```

WHERE id = p_id;

RETURN proveedor;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20035, 'Error al obtener proveedor');

END obtener_proveedor;


FUNCTION obtener_compra_productos RETURN SYS_REFCURSOR AS --10
    compra_productos SYS_REFCURSOR;

BEGIN

    OPEN compra_productos FOR

    SELECT * FROM COMPRA_PRODUCTO

    INNER JOIN COMPRA_DETALLE

    ON COMPRA_PRODUCTO.id = COMPRA_DETALLE.compra_producto_id;

    RETURN compra_productos;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20036, 'Error al obtener compra productos');

END obtener_compra_productos;


FUNCTION obtener_compra_producto(p_id NUMBER) RETURN SYS_REFCURSOR AS --11
    compra_producto SYS_REFCURSOR;

BEGIN

    OPEN compra_producto FOR

    SELECT * FROM COMPRA_PRODUCTO

    INNER JOIN COMPRA_DETALLE

    ON COMPRA_PRODUCTO.id = COMPRA_DETALLE.compra_producto_id

    WHERE COMPRA_PRODUCTO.id = p_id;

    RETURN compra_producto;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20037, 'Error al obtener compra producto');

END obtener_compra_producto;

```



```

FUNCTION obtener_articulos_vendido_mes(anio_mes DATE) RETURN NUMBER AS
    articulos_vendidos NUMBER;
BEGIN
    SELECT SUM(cantidad)
    INTO articulos_vendidos
    FROM PRODUCTO_RECIBO
    INNER JOIN RECIBO
    ON PRODUCTO_RECIBO.recibo_id = RECIBO.id
    WHERE EXTRACT(MONTH FROM RECIBO.fecha) = EXTRACT(MONTH FROM anio_mes) AND EXTRACT(YEAR
    FROM RECIBO.fecha) = EXTRACT(YEAR FROM anio_mes);

    IF articulos_vendidos IS NULL THEN
        articulos_vendidos := 0;
    END IF;

    RETURN articulos_vendidos;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20038, 'Error al obtener articulos vendidos');
END obtener_articulos_vendido_mes;

```

```

FUNCTION obtener_articulo_vendido_mes(p_id NUMBER, anio_mes DATE) RETURN NUMBER AS
    articulo_vendido NUMBER;
BEGIN
    SELECT SUM(cantidad)
    INTO articulo_vendido
    FROM PRODUCTO_RECIBO
    INNER JOIN RECIBO
    ON PRODUCTO_RECIBO.recibo_id = RECIBO.id
    WHERE EXTRACT(MONTH FROM RECIBO.fecha) = EXTRACT(MONTH FROM anio_mes) AND EXTRACT(YEAR
    FROM RECIBO.fecha) = EXTRACT(YEAR FROM anio_mes) AND PRODUCTO_RECIBO.producto_id = p_id;

    IF (articulo_vendido IS NULL) THEN
        articulo_vendido := 0;
    END IF;

    RETURN articulo_vendido;

```

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20039, 'Error al obtener articulo vendido');

END obtener_articulo_vendido_mes;

FUNCTION obtener_producto_recibo_todos RETURN SYS_REFCURSOR AS

producto_recibo SYS_REFCURSOR;

BEGIN

OPEN producto_recibo FOR

SELECT * FROM PRODUCTO_RECIBO;

RETURN producto_recibo;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20040, 'Error al obtener producto recibo');

END obtener_producto_recibo_todos;

FUNCTION obtener_producto_recibo(p_producto_id NUMBER, p_recibo_id NUMBER) RETURN
SYS_REFCURSOR AS

producto_recibo SYS_REFCURSOR;

BEGIN

OPEN producto_recibo FOR

SELECT * FROM PRODUCTO_RECIBO

WHERE producto_id = p_producto_id AND recibo_id = p_recibo_id;

RETURN producto_recibo;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20041, 'Error al obtener producto recibo');

END obtener_producto_recibo;

FUNCTION obtener_productos_recibo(p_recibo_id NUMBER) RETURN SYS_REFCURSOR AS

productos_recibo SYS_REFCURSOR;

BEGIN

OPEN productos_recibo FOR

```

SELECT * FROM PRODUCTO_RECIBO

WHERE recibo_id = p_recibo_id;

RETURN productos_recibo;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20042, 'Error al obtener productos recibo');

END obtener_productos_recibo;


FUNCTION obtener_producto_recibos(p_producto_id NUMBER) RETURN SYS_REFCURSOR AS

    producto_recibos SYS_REFCURSOR;

BEGIN

    OPEN producto_recibos FOR

    SELECT * FROM PRODUCTO_RECIBO

    WHERE producto_id = p_producto_id;

    RETURN producto_recibos;

EXCEPTION

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20043, 'Error al obtener producto recibos');

END obtener_producto_recibos;


END paquete_select;

/


-- INSERTS


BEGIN

    INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Pedro Perez', DATE '2023-05-05', 'pedro@gmail.com',
    'Grecia Centro', 79);

    INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Juan Paez', DATE '2023-05-05', 'juan@gmail.com',
    'Grecia Centro', 69);

    INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Julio Asir', DATE '2023-05-07', 'julio@gmail.com',
    'Grecia Centro', 89);

    INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Vardimir Yusuf', DATE '2023-02-05', 'tusuf@gmail.com',
    'Grecia Centro', 10);

```

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Isaias Reque', DATE '2023-05-01', 'reque@gmail.com', 'Grecia Centro', 99);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Manuel Calderon', DATE '2023-03-05', 'manuel@gmail.com', 'Grecia Centro', 69);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Melvin Ramirez', DATE '2023-05-09', 'mel@gmail.com', 'Grecia Centro', 10);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Koky Saldo', DATE '2023-02-05', 'koky@gmail.com', 'Grecia Centro', 12);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Probabilidad Fernandez', DATE '2023-05-05', 'proba@gmail.com', 'Grecia Centro', 80);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Fundamentos Charpentier', DATE '2023-04-05', 'funda@gmail.com', 'Grecia Centro', 56);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Arquitectura Gomez', DATE '2023-05-08', 'arqui@gmail.com', 'Grecia Centro', 23);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Tzu Rue', DATE '2023-09-05', 'tzurue@gmail.com', 'Grecia Centro', 12);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Algebra Gonzales', DATE '2023-05-01', 'alge@gmail.com', 'Grecia Centro', 90);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Discreta Alvarez', DATE '2023-05-01', 'discreta@gmail.com', 'Grecia Centro', 67);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Volador Enriquez', DATE '2023-08-02', 'volador@gmail.com', 'Grecia Centro', 11);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Manrique Lopez', DATE '2023-01-01', 'manriquito@gmail.com', 'Grecia Centro', 78);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Kosta Latysh', DATE '2023-03-03', 'kosta@gmail.com', 'Grecia Centro', 12);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Yargen Humberto', DATE '2023-05-01', 'yargen@gmail.com', 'Grecia Centro', 44);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Leonardo Hulio', DATE '2023-02-02', 'leonardo@gmail.com', 'Grecia Centro', 90);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Erling Lopez', DATE '2023-02-05', 'erling@gmail.com', 'Grecia Centro', 290);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Manfred Messi', DATE '2023-05-01', 'godfred@gmail.com', 'Grecia Centro', 80);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Alejandro Ronaldo', DATE '2023-09-01', 'ale@gmail.com', 'Grecia Centro', 71);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Federico Roman', DATE '2023-05-09', 'fede@gmail.com', 'Grecia Centro', 123);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Julian Nuñez', DATE '2023-02-08', 'julia@gmail.com', 'Grecia Centro', 6);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Maria Messi', DATE '2023-01-05', 'maria@gmail.com', 'Grecia Centro', 100);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Danna Porras', DATE '2023-05-01', 'danna@gmail.com', 'Grecia Centro', 12);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Bad Bunny', DATE '2023-03-05', 'chambea@gmail.com', 'Grecia Centro', 90);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Juan Balvin', DATE '2023-02-05', 'jbalvin@gmail.com', 'Grecia Centro', 4);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Maluma Baby', DATE '2023-02-09', 'maluma@gmail.com', 'Grecia Centro', 69);

INSERT INTO CLIENTE VALUES (seq_cliente.NEXTVAL, 'Maripaz Cespedes', DATE '2023-02-05', 'koky@gmail.com', 'Grecia Centro', 55);

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Quesos Rubi', 'Trae quesos', 'Alajuela centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Carnes San Jose', 'Trae carnes', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Electronicos Luisito', 'Trae aparatos electronicos', 'Grecia, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Papas El carton', 'Proveedor de papas', 'Grecia, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Carnes San Carlos', 'Trae carnes de San Carlos', 'San Carlos, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Leche San Osito', 'Trae lacteos', 'Cartago centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Arroces el Burro', 'Proveedor de arroz', 'Puntarenas centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Frijolera Ruiz', 'Proveedor de frijoles', 'San Ramon, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Mantequillas San Humberto', 'Trae mantequillas', 'Barva, Heredia');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Tortillera El maicito', 'Proveedor de tortillas', 'San Carlos, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Especias Marito', 'Proveedor de especias', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Filtros el Hugo', 'Trae filtros para cafeteras', 'Guanacaste');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Chapstick Life', 'Proveedor de chapsticks', 'Alajuela, centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Lapiceros Manuelito', 'Proveedor de lapiceros', 'Escazú, San José');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Fundas CeluWorld', 'Proveedor de fundas para telefono', 'Heredia centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Frescos La hortaliza', 'Trae refrescos variados', 'San Carlos, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Cafes el grano', 'Proveedor de cafes', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Cafes el Cafe', 'Proveedor de cafes', 'Pavas, San José');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Grapas el grapon', 'Proveedor de grapas', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Borradores Lucho', 'Proveedor de borradores', 'Santo Domingo, Heredia');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Paquetillos el enamorado', 'Proveedor de de tostadospaquetillos ', 'San Luis, Grecia');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Caramelos el dulce', 'Proveedor de caramelos', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Confites el Ruso', 'Proveedor de confites rusos', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Porcelana la Porcelana fina', 'Proveedor de porcelana', 'Leon 13');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Relojos el Reoljito', 'Proveedor de relojes', 'San Ramon, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Yucas la Fortuna', 'Proveedor de caramelos', 'La fortuna');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Paperia las papitas', 'Proveedor de papas', 'Cartago');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Carnes la Chuleteria', 'Trae chuketas', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'Pipo Pages', 'Trae hojas', 'San Jose centro');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'El manjar del perro', 'Trae comida de perros', 'Grecia, Alajuela');

INSERT INTO PROVEEDOR VALUES (seq_proveedor.NEXTVAL, 'El manjar del gato', 'Trae comida de gatos', 'San Jose centro');

INSERT INTO CATEGORIA VALUES ('Electrónica', 'Productos electrónicos para el consumo diario.');

INSERT INTO CATEGORIA VALUES ('Ropa', 'Ropa y accesorios de moda para hombres y mujeres.');

INSERT INTO CATEGORIA VALUES ('Calzado', 'Calzado de diversos estilos y tallas.');

INSERT INTO CATEGORIA VALUES ('Electrodomésticos', 'Electrodomésticos para el hogar');

INSERT INTO CATEGORIA VALUES ('Joyería', 'Joyas y bisutería');

```

INSERT INTO CATEGORIA VALUES ('Muebles', 'Muebles y decoración para el hogar.');
```

```

INSERT INTO CATEGORIA VALUES ('Productos de Belleza', 'Productos de cuidado personal y belleza.');
```

```

INSERT INTO CATEGORIA VALUES ('Juguetes', 'Juguetes para niños.');
```

```

INSERT INTO CATEGORIA VALUES ('Librería', 'Productos para estudio u oficina.');
```

```

INSERT INTO CATEGORIA VALUES ('Deportivo', 'Artículos para deporte');
```

```

INSERT INTO CATEGORIA VALUES ('Carnes', 'Productos hechos de carne');
```

```

INSERT INTO CATEGORIA VALUES ('Bebés', 'Productos para bebés y cuidado infantil.');
```

```

INSERT INTO CATEGORIA VALUES ('Bebidas', 'Productos que se consumen tomándolo');
```

```

INSERT INTO CATEGORIA VALUES ('Ferretería', 'Herramientas y suministros de ferretería.');
```

```

INSERT INTO CATEGORIA VALUES ('Mariscos', 'Productos provenientes del mar');
```

```

INSERT INTO CATEGORIA VALUES ('Limpieza', 'Productos de limpieza');
```

```

INSERT INTO CATEGORIA VALUES ('Snacks', 'Productos para picar');
```

```

INSERT INTO CATEGORIA VALUES ('Jardinería', 'Suministros y herramientas para jardinería.');
```

```

INSERT INTO CATEGORIA VALUES ('Mascotas', 'Suministros y alimentos para mascotas.');
```

```

INSERT INTO CATEGORIA VALUES ('Congelados', 'Productos congelados');
```

```

INSERT INTO CATEGORIA VALUES ('Instrumentos Musicales', 'Instrumentos musicales y accesorios.');
```

```

INSERT INTO CATEGORIA VALUES ('Sexo', 'Productos relacionado al coito');
```

```

INSERT INTO CATEGORIA VALUES ('Cocina', 'Utensilios y productos para cocina.');
```

```

INSERT INTO CATEGORIA VALUES ('Dulces', 'Confites, chocolates, etc...');
```

```

INSERT INTO CATEGORIA VALUES ('Fiesta', 'Suministros para fiestas y eventos especiales.');
```

```

INSERT INTO CATEGORIA VALUES ('Embutidos', 'Productos embutidos');
```

```

INSERT INTO CATEGORIA VALUES ('Frutas', 'Frutas para la familia');
```

```

INSERT INTO CATEGORIA VALUES ('Vegetales', 'Verduras para la familia');
```

```

INSERT INTO CATEGORIA VALUES ('Enlatados', 'Productos que vienen en una lata');
```

```

INSERT INTO CATEGORIA VALUES ('Lácteos', 'Productos provenientes de la leche');
```



```

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Samsung 78', 'Teléfono móvil inteligente', 250000, 'Electrónica', 15);
```

```

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Vestido Shein', 'Elegante vestido de noche', 15000, 'Ropa', 32);
```

```

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Naik', 'Calzado deportivo para correr y hacer ejercicio', 35000, 'Calzado', 5);
```

```

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Refrigerador Samsung', 'Refrigerador de gran capacidad', 650000, 'Electrodomésticos', 3);
```

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Anillo de plata', 'Anillo con plata de alta calidad', 80000, 'Joyería', 2);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Sofá', 'Sofá de cuero de lujo', 350000, 'Muebles', 1);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Crema Hidratante', 'Crema hidratante para el cuidado de la piel', 13000, 'Productos de Belleza', 67);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Blocks', 'Set de construcción de bloques para niños', 10000, 'Juguetes', 43);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Cuaderno de Notas', 'Cuaderno de notas para tomar apuntes', 3000, 'Librería', 13);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Balón de Fútbol', 'Balón de fútbol oficial', 20000, 'Deportivo', 17);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Filete de Res', 'Filete de res de primera calidad', 2000, 'Carnes', 120);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Pañales para Bebés', 'Pañales desechables para bebés', 24000, 'Bebés', 23);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Vino Tinto', 'Botella de vino tinto de alta calidad', 50000, 'Bebidas', 20);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Martillo', 'Herramienta para clavar', 7000, 'Ferretería', 100);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Camarones Frescos', 'Camarones frescos para cocinar', 16000, 'Mariscos', 30);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Detergente Líquido', 'Detergente líquido para lavar ropa', 5000, 'Limpieza', 14);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Papiola', 'Chips de papas crujientes y sabrosos', 300, 'Snacks', 300);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Tijeras', 'Tijeras para jardín', 4500, 'Jardinería', 12);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Alimento para Gatos', 'Alimento de alta calidad para gatos', 3600, 'Mascotas', 25);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Pizza Congelada', 'Pizza congelada para una cena rápida', 3000, 'Congelados', 60);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Flauta', 'Flauta para músicos aficionados', 15000, 'Instrumentos Musicales', 15);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Condomes', 'Condomes de calidad para la protección', 2500, 'Sexo', 200);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Cuchillo', 'Cuchillo para cocinar', 5500, 'Cocina', 40);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Barra de Chocolate', 'Barra de chocolate suizo premium', 1500, 'Dulces', 25);


```
INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Globos', 'Globos para fiestas', 2500, 'Fiesta', 26);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Salchichas Ahumadas', 'Salchichas ahumadas para parrilladas', 4500, 'Embutidos', 50);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Manzanas', 'Manzanas frescas y crujientes', 1000, 'Frutas', 200);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Zanahorias', 'Zanahorias orgánicas', 1250, 'Vegetales', 96);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Atún', 'Atún de agua', 400, 'Enlatados', 56);

INSERT INTO PRODUCTO VALUES (seq_producto.NEXTVAL, 'Leche Entera', 'Leche entera fresca y nutritiva', 1700, 'Lácteos', 80);
```

```
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,1,1,DATE '2023-20-09');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,1,1,DATE '2023-20-10');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,1,1,DATE '2023-20-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,2,2,DATE '2023-20-12');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,2,2,DATE '2023-07-13');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,2,2,DATE '2023-08-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,3,3,DATE '2023-20-09');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,3,3,DATE '2023-20-10');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,3,3,DATE '2023-20-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,4,4,DATE '2023-20-12');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,4,4,DATE '2023-07-13');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,4,4,DATE '2023-08-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,5,5,DATE '2023-20-09');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,5,5,DATE '2023-20-10');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,5,5,DATE '2023-20-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,6,6,DATE '2023-20-12');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,6,6,DATE '2023-07-13');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,6,6,DATE '2023-08-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,7,7,DATE '2023-20-09');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,7,7,DATE '2023-20-10');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,7,7,DATE '2023-20-11');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,8,8,DATE '2023-20-12');
INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,8,8,DATE '2023-07-13');
```

[illegible]

[illegible]

INSERT INTO COMPRA_PRODUCTO VALUES(seq_compra_producto.NEXTVAL,30,30,DATE '2023-08-11');

INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 1, 10, 2000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 2, 15, 3000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 3, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 4, 12, 2400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 5, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 6, 20, 4000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 7, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 8, 18, 3600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 9, 10, 2000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 10, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 11, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 12, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 13, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 14, 14, 2800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 15, 11, 2200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 16, 2, 400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 17, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 18, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 19, 16, 3200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 20, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 21, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 22, 12, 2400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 23, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 24, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 25, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 26, 13, 2600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 27, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 28, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 29, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 30, 11, 2200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 1, 10, 2000, 'Productos variados');

```
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 2, 15, 3000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 3, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 4, 12, 2400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 5, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 6, 20, 4000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 7, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 8, 18, 3600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 9, 10, 2000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 10, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 11, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 12, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 13, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 14, 14, 2800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 15, 11, 2200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 16, 2, 400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 17, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 18, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 19, 16, 3200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 20, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 21, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 22, 12, 2400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 23, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 24, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 25, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 26, 13, 2600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 27, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 28, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 29, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 30, 11, 2200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 1, 10, 2000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 2, 15, 3000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 3, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 4, 12, 2400, 'Productos variados');
```

```
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 5, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 6, 20, 4000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 7, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 8, 18, 3600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 9, 10, 2000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 10, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 11, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 12, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 13, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 14, 14, 2800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 15, 11, 2200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 16, 2, 400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 17, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 18, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 19, 16, 3200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 20, 6, 1200, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 21, 3, 600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 22, 12, 2400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 23, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 24, 4, 800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 25, 7, 1400, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 26, 13, 2600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 27, 8, 1600, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 28, 5, 1000, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 29, 9, 1800, 'Productos variados');
INSERT INTO COMPRA_DETALLE VALUES (seq_compra_detalle.NEXTVAL, 30, 11, 2200, 'Productos variados');
```

```
INSERT INTO USUARIO(nombre, passhash1, passhash2, nivel) VALUES('admin', 133524947, 894447882, 1);
```

```
COMMIT;
```

```
END;
```

```
DROP TRIGGER bitacora_producto_insert;  
DROP TRIGGER bitacora_producto_update;  
DROP TRIGGER bitacora_producto_delete;  
DROP TRIGGER recibo_insert;
```

```
DROP PACKAGE paquete_modificar;  
DROP PACKAGE paquete_select;
```

```
DROP SEQUENCE seq_bitacora_producto;  
DROP SEQUENCE seq_compra_detalle;  
DROP SEQUENCE seq_compra_producto;  
DROP SEQUENCE seq_proveedor;  
DROP SEQUENCE seq_recibo_detalle;  
DROP SEQUENCE seq_recibo;  
DROP SEQUENCE seq_cliente;  
DROP SEQUENCE seq_producto;
```

```
DROP TABLE BITACORA_PRODUCTO;  
DROP TABLE COMPRA_DETALLE;  
DROP TABLE COMPRA_PRODUCTO;  
DROP TABLE PROVEEDOR;  
DROP TABLE RECIBO_DETALLE;  
DROP TABLE PRODUCTO_RECIBO;  
DROP TABLE RECIBO;  
DROP TABLE CLIENTE;  
DROP TABLE PRODUCTO;  
DROP TABLE CATEGORIA;  
DROP TABLE USUARIO;
```