Proyecto I (Aplicación Web de apoyo para las PYMES)

Profesor

Alberto Shum Chan

Bases de datos II

Integrantes

Taylor Hernandez (2020196104) y Pablo Muñoz (2020031899)

ITCR

SEP 16, 2021

```sql
-- TABLAS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

DROP TABLE Linea_de_producto;
DROP TABLE Orden;
DROP TABLE Registro;
DROP TABLE Producto;
DROP TABLE Categoria;
DROP TABLE Cliente;
DROP TABLE Pyme;

CREATE TABLE Cliente (
   cliente_id NUMBER GENERATED by default on null as IDENTITY,
   cedula     VARCHAR2(12),
   nombre_cliente     VARCHAR2(45),
   apodo  VARCHAR2(25),
   apellido1  VARCHAR2(25),
   apellido2  VARCHAR2(25),
   telefono_cliente   VARCHAR2(15),
   contraseña   BLOB,
   provincia_cliente   VARCHAR2(20),
   PRIMARY KEY ( cliente_id )
);

CREATE TABLE Pyme (
   pyme_id       NUMBER GENERATED by default on null as IDENTITY,
   cedula_juridica     VARCHAR2(12),
   nombre_pyme     VARCHAR2(45),
   email         VARCHAR2(85),
   telefono_pyme   VARCHAR2(15),
   clave   BLOB,
   provincia_pyme   VARCHAR2(20),
   PRIMARY KEY ( pyme_id )
);

CREATE TABLE Categoria (
   categoria_id NUMBER GENERATED by default on null as IDENTITY,
   nombre_categoria     VARCHAR2(100),
   PRIMARY KEY ( categoria_id )
);

CREATE TABLE Producto (
   producto_id NUMBER GENERATED by default on null as IDENTITY,
   nombre_producto     VARCHAR2(55),
   descripcion  VARCHAR2(255),
   precio_unitario  NUMBER(10,2),
   categoria_id   NUMBER,
   pyme_id    NUMBER,
   PRIMARY KEY ( producto_id ),
   CONSTRAINT fk_categoria_producto
      FOREIGN KEY (categoria_id)
      REFERENCES Categoria (categoria_id),
   CONSTRAINT fk_pyme_producto
      FOREIGN KEY (pyme_id)
      REFERENCES Pyme (pyme_id)
);

CREATE TABLE Registro (
   registro_id NUMBER GENERATED by default on null as IDENTITY,
   monto_anterior     NUMBER(10,2),
   fecha_registro  DATE,
   producto_id  NUMBER,
   PRIMARY KEY ( registro_id ),
   CONSTRAINT fk_producto_registro
      FOREIGN KEY (producto_id)
      REFERENCES Producto (producto_id)
);

CREATE TABLE Orden (
```

```
    orden_id NUMBER GENERATED by default on null as IDENTITY,
    monto_total      NUMBER(10,2),
    estado   NUMBER(1),
    fecha_orden   DATE,
    cliente_id   NUMBER,
    PRIMARY KEY ( orden_id ),
    CONSTRAINT fk_cliente_orden
        FOREIGN KEY (cliente_id)
        REFERENCES Cliente (cliente_id)
);

CREATE TABLE Linea_de_producto (
    linea_id NUMBER GENERATED by default on null as IDENTITY,
    cantidad      NUMBER,
    monto   NUMBER(10,2),
    orden_id   NUMBER,
    producto_id   NUMBER,
    PRIMARY KEY ( linea_id ),
    CONSTRAINT fk_orden
        FOREIGN KEY (orden_id)
        REFERENCES Orden (orden_id),
    CONSTRAINT fk_producto_linea
        FOREIGN KEY (producto_id)
        REFERENCES Producto (producto_id)
);
```

```
insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseña, provincia_cliente) values ('1-1815-0121', 'Pablo', 'Munoz', 'Hidalgo', 'Puvlo', '8431-0740', utl_raw.cast_to_raw('PMH'), 'Guanacaste');
insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseña, provincia_cliente) values ('2-4415-5515', 'Taylor', 'Hernandez', 'Cordoba', 'Santi01', '8465-1302', utl_raw.cast_to_raw('THC'), 'Puntarenas');
insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseña, provincia_cliente) values ('3-5541-1545', 'Alberto', 'Shum', 'Chan', 'ShumChany10', '8745-6212', utl_raw.cast_to_raw('ASC'), 'Heredia');
insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseña, provincia_cliente) values ('9-1545-5465', 'Jaime', 'Rojas', 'Ramirez', 'Jaimito2519', '4545-4554', utl_raw.cast_to_raw('JRR'), 'Alajuela');
insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseña, provincia_cliente) values ('8-7982-5648', 'Deyton', 'Hernandez', 'Cordoba', 'Patas', '8411-2257', utl_raw.cast_to_raw('DHC'), 'Limon');

insert into Categoria(nombre_categoria) values ('Cocina');
insert into Categoria(nombre_categoria) values ('Tecnologia');
insert into Categoria(nombre_categoria) values ('Ropa');
insert into Categoria(nombre_categoria) values ('Placer');
insert into Categoria(nombre_categoria) values ('Carnes');

insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values ('3-515-5448', 'AstroFeelings', 'afeeling@gmail.com', '5715-2548', utl_raw.cast_to_raw('ronaldouno'), 'San Jose');
insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values ('3-515-5415', 'TechoSpace', 'technospace@gmail.com', '5744-2548', utl_raw.cast_to_raw('ronaldodos'), 'Alajuela');
insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values ('3-515-2101', 'PrimeBeef', 'primebeef@gmail.com', '5754-2178', utl_raw.cast_to_raw('ronaldotres'), 'San Jose');
insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values ('3-515-8452', 'Shoelaces', 'shoelaces@gmail.com', '5715-2542', utl_raw.cast_to_raw('ronaldocuatro'), 'Guanacaste');
insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values ('3-515-1564', 'Homeandcook', 'handc@gmail.com', '5715-2841', utl_raw.cast_to_raw('ronaldocinco'), 'San Jose');

insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values ('Smartphone Samsung S5', 'Telefono movil para uso diario', 550000, 2, 2);
insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values ('Camisa Nike', 'Logo en la parte frontal de color negro con blanco', 15000, 3, 4);
insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values ('Esposas SexyTime', 'Esposas acolchonadas de color rosa', 10500, 4, 1);
insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values ('Guantes de cocina C8200', 'Guantes de cocina para evitar el calor', 17500, 1, 5);
insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values ('Rib-Eye Premium USA Prime El Arreo', 'Carne premium el arreo Rib-Eye para asadito con los panas', 45000, 5, 3);

insert into Orden( monto_total, estado, fecha_orden, cliente_id) values (225522, 1, '02-DEC-1997', 2);
insert into Orden( monto_total, estado, fecha_orden, cliente_id) values (455646, 0, '27-NOV-2001', 3);
insert into Orden( monto_total, estado, fecha_orden, cliente_id) values (515465.50, 1, '21-OCT-2008', 4);
insert into Orden( monto_total, estado, fecha_orden, cliente_id) values (564874, 0, '29-OCT-2010', 1);
insert into Orden( monto_total, estado, fecha_orden, cliente_id) values (979879, 1, '12-JAN-2018', 5);

insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (1, 550000, 1, 1);
insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (2, 30000, 2, 2);
insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (69, 724500, 3, 3);
insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (2, 35000, 4, 4);
insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (10, 450000, 5, 5);

insert into Registro( monto_anterior, fecha_registro, producto_id) values (250000, '05-AUG-1995', 1);
insert into Registro( monto_anterior, fecha_registro, producto_id) values (2500, '08-SEP-1990', 1);
insert into Registro( monto_anterior, fecha_registro, producto_id) values (15000, '09-JUN-1998', 3);
insert into Registro( monto_anterior, fecha_registro, producto_id) values (10000, '07-JUL-2010', 4);
insert into Registro( monto_anterior, fecha_registro, producto_id) values (40000, '01-FEB-2020', 5);

commit;
```

```
-- ================================================================================

--                          CRUD

-- ================================================================================
--                          ORDEN
-- ================================================================================

create or replace procedure orden_insert (
new_monto_total NUMBER,
new_estado   NUMBER,
new_fecha_orden   DATE,
new_cliente_id   NUMBER
) AS

begin
   insert into Orden( monto_total, estado, fecha_orden, cliente_id) values ( new_monto_total, new_estado, new_fecha_orden,
new_cliente_id);
end;
/

create or replace procedure orden_update (
new_orden_id NUMBER,
new_monto_total NUMBER,
new_estado   NUMBER,
new_fecha_orden   DATE,
new_cliente_id   NUMBER
) AS

begin
   update Orden set monto_total = new_monto_total, estado = new_estado, fecha_orden = new_fecha_orden, cliente_id =
new_cliente_id
   where orden_id = new_orden_id;
end;
/

create or replace procedure orden_delete (
new_orden_id NUMBER
) AS
begin
   delete from Orden where orden_id = new_orden_id;
end;
/


-- ================================================================================
--                          Cliente
-- ================================================================================
create or replace procedure cliente_insert (
   new_cedula      VARCHAR2,
   new_nombre_cliente      VARCHAR2,
   new_apellido1   VARCHAR2,
   new_apellido2   VARCHAR2,
   new_apodo   VARCHAR2,
   new_telefono_cliente    VARCHAR2,
   new_contraseńa    VARCHAR2,
   new_provincia_cliente    VARCHAR2
   ) AS
   new_pass raw(50);
begin
   new_pass:= utl_raw.cast_to_raw(new_contraseńa);
   insert into Cliente(cedula, nombre_cliente, apellido1, apellido2, apodo, telefono_cliente, contraseńa, provincia_cliente)
values ( new_cedula, new_nombre_cliente, new_apellido1, new_apellido2, new_apodo, new_telefono_cliente, new_pass,
new_provincia_cliente);
end;
/
```

```
create or replace procedure cliente_update (
   old_id NUMBER,
   new_cliente_id NUMBER,
   new_cedula      VARCHAR2,
   new_nombre_cliente      VARCHAR2,
   new_apellido1   VARCHAR2,
   new_apellido2   VARCHAR2,
   new_apodo   VARCHAR2,
   new_telefono_cliente    VARCHAR2,
   new_contraseńa    VARCHAR2,
   new_provincia_cliente    VARCHAR2
   )  AS
   new_pass raw(50);
begin
   new_pass:= utl_raw.cast_to_raw(new_contraseńa);
   update Cliente set cliente_id = new_cliente_id, cedula = new_cedula, nombre_cliente = new_nombre_cliente, apellido1 =
new_apellido1, apellido2 = new_apellido2, apodo = new_apodo, telefono_cliente = new_telefono_cliente , contraseńa =
new_pass, provincia_cliente = new_provincia_cliente
   where cliente_id = old_id;
end;
/


create or replace procedure cliente_delete (
   new_cliente_id NUMBER
   ) AS
begin
   delete from Cliente where cliente_id = new_cliente_id;
end;
/

-- ============================================================================
--                          Producto
-- ============================================================================
create or replace procedure producto_insert (
   new_nombre_producto      VARCHAR2,
   new_descripcion   VARCHAR2,
   new_precio_unitario   NUMBER,
   new_categoria_id    NUMBER,
   new_pyme_id     NUMBER
   ) AS

begin
   insert into Producto( nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values (new_nombre_producto,
new_descripcion, new_precio_unitario, new_categoria_id, new_pyme_id);
end;
/

create or replace procedure producto_update (
   new_producto_id NUMBER,
   new_nombre_producto      VARCHAR2,
   new_descripcion   VARCHAR2,
   new_precio_unitario   NUMBER,
   new_categoria_id    NUMBER,
   new_pyme_id     NUMBER
   ) AS

begin
   update Producto set nombre_producto  = new_nombre_producto, descripcion = new_descripcion, precio_unitario =
new_precio_unitario, categoria_id = new_categoria_id, pyme_id = new_pyme_id
   where producto_id = new_producto_id;
end;
/

create or replace procedure producto_update_precio (
   new_producto_id NUMBER,
```

```
    new_precio_unitario   NUMBER
    ) AS

begin
    update Producto set precio_unitario = new_precio_unitario
    where producto_id = new_producto_id;
end;
/

create or replace procedure delete_producto (
    new_producto_id NUMBER
    ) AS
begin
    delete from Producto where producto_id = new_producto_id;
end;
/

-- ========================================================================
--                          Pyme
-- ========================================================================
create or replace procedure pyme_insert (
    new_cedula_juridica      VARCHAR2,
    new_nombre_pyme      VARCHAR2,
    new_email          VARCHAR2,
    new_telefono_pyme    VARCHAR2,
    new_clave    VARCHAR2,
    new_provincia_pyme    VARCHAR2
    )  AS
    new_pass raw(50);
begin
    new_pass:= utl_raw.cast_to_raw(new_clave);
    insert into Pyme( cedula_juridica, nombre_pyme, email, telefono_pyme, clave, provincia_pyme) values (
new_cedula_juridica, new_nombre_pyme, new_email, new_telefono_pyme, new_pass, new_provincia_pyme);
end;
/

create or replace procedure update_pyme (
    old_id NUMBER,
    new_pyme_id        NUMBER,
    new_cedula_juridica      VARCHAR2,
    new_nombre_pyme      VARCHAR2,
    new_email          VARCHAR2,
    new_telefono_pyme    VARCHAR2,
    new_clave    VARCHAR2,
    new_provincia_pyme    VARCHAR2
    ) AS
    new_pass raw(50);
begin
    new_pass:= utl_raw.cast_to_raw(new_clave);
    update Pyme set pyme_id = new_pyme_id, cedula_juridica = new_cedula_juridica, nombre_pyme = new_nombre_pyme,
email = new_email, telefono_pyme = new_telefono_pyme, clave = new_pass, provincia_pyme = new_provincia_pyme
    where pyme_id=old_id;
end;
/

create or replace procedure pyme_delete (
    new_pyme_id        NUMBER
    ) AS
begin
    delete from Pyme where pyme_id = new_pyme_id;
end;
/

-- ========================================================================
--                          Categoria
```

```
-- ============================================================================
create or replace procedure categoria_insert  (
   new_nombre_categoria      VARCHAR2
   ) AS

begin
   insert into Categoria( nombre_categoria) values ( new_nombre_categoria);
end;
/

create or replace procedure categoria_update   (
   new_categoria_id NUMBER,
   new_nombre_categoria      VARCHAR2
   ) AS
begin
   update Categoria set nombre_categoria = new_nombre_categoria
   where categoria_id = new_categoria_id;
end;
/

create or replace procedure categoria_delete as
new_categoria_id number;
begin
   delete from Categoria where categoria_id = new_categoria_id;
end;
/

-- ============================================================================
--                          Linea de producto
-- ============================================================================
create or replace procedure linea_de_producto_insert (
   new_cantidad      NUMBER,
   new_monto   NUMBER,
   new_orden_id   NUMBER,
   new_producto_id   NUMBER
   ) AS

begin
   insert into Linea_de_producto( cantidad, monto, orden_id, producto_id) values (new_cantidad, new_monto, new_orden_id,
new_producto_id);
end;
/

create or replace procedure linea_de_producto_update (
   new_linea_id NUMBER,
   new_cantidad      NUMBER,
   new_monto   NUMBER,
   new_orden_id   NUMBER,
   new_producto_id   NUMBER
   ) AS

begin
   update Linea_de_producto set cantidad = new_cantidad, monto = new_monto, orden_id = new_orden_id, producto_id =
new_producto_id
   where linea_id = new_linea_id;
end;
/

create or replace procedure linea_de_producto_delete as
   new_linea_id number;
begin
   delete from Linea_de_producto where linea_id = new_linea_id;
end;
/
```

```
-- ============================================================================
--                          Registro
-- ============================================================================
create or replace procedure registro_insert (
   new_monto_anterior      NUMBER,
   new_fecha_registro   DATE,
   new_producto_id   NUMBER
   ) AS

begin
   insert into Registro( monto_anterior, fecha_registro, producto_id) values ( new_monto_anterior, new_fecha_registro,
new_producto_id);
end;
/

create or replace procedure registro_update (
   new_registro_id NUMBER,
   new_monto_anterior      NUMBER,
   new_fecha_registro   DATE,
   new_producto_id   NUMBER
   ) AS
begin
   update Registro set monto_anterior = new_monto_anterior, fecha_registro = new_fecha_registro, producto_id =
new_producto_id
   where registro_id = new_registro_id;
end;
/

create or replace procedure registro_delete(
   new_registro_id NUMBER
   ) AS
begin
   delete from Registro where registro_id = new_registro_id;
end;
/


commit;
```

```
-- VALIDATION OF THE CLIENT DATA
create or replace FUNCTION FC_VALID_CLIENTE (
   new_cedula      VARCHAR2,
   new_apodo   VARCHAR2,
   new_telefono_cliente    VARCHAR2
   )
RETURN NUMBER
IS
   apodex NUMBER;
   cedulex NUMBER;
   numberx NUMBER;
begin

   SELECT COUNT(*) INTO apodex from Cliente where new_apodo=apodo;
   SELECT COUNT(*) INTO cedulex from Cliente where new_cedula=cedula;
   SELECT COUNT(*) INTO numberx from Cliente where new_telefono_cliente=telefono_cliente;

   IF apodex=0 and cedulex=0 and numberx=0 then
      return 0;
   Else
      IF apodex!=0 then
         return 1;
      END IF;

      IF cedulex!=0 then
         return 2;
      END IF;

      IF numberx!=0 then
         return 3;
      END IF;
   end if;
end;
/

-- VALIDATION OF THE PYMES DATA
create or replace FUNCTION FC_VALID_PYME (
   new_cedula      VARCHAR2,
   new_email   VARCHAR2,
   new_telefono_cliente    VARCHAR2
   )
RETURN NUMBER
IS
   emailex NUMBER;
   cedulex NUMBER;
   numberx NUMBER;
begin

   SELECT COUNT(*) INTO emailex from Pyme where new_email=email;
   SELECT COUNT(*) INTO cedulex from Pyme where new_cedula=cedula_juridica;
   SELECT COUNT(*) INTO numberx from Pyme where new_telefono_cliente=telefono_pyme;

   IF emailex=0 and cedulex=0 and numberx=0 then
      return 0;
   Else
      IF emailex!=0 then
         return 1;
      END IF;

      IF cedulex!=0 then
         return 2;
      END IF;

      IF numberx!=0 then
         return 3;
```

```
        END IF;
    end if;
end;
/

commit;
```

```
-- LOGIN DEL CLIENTE
CREATE OR REPLACE FUNCTION FC_CLIENTE_LOGIN(ident VARCHAR2, pass VARCHAR2)
RETURN number is
resultado number;
c_id number;
new_pass BLOB;
begin
   new_pass:= TO_BLOB(utl_raw.cast_to_raw(pass));
   select count(*) into resultado from Cliente cl where (ident=cl.apodo or  ident=cl.telefono_cliente or ident=cl.cedula) and
dbms_lob.compare(new_pass, cl.contraseńa) = 0;
   if resultado = 1 then
      select cliente_id into c_id from Cliente cl where (ident=cl.apodo or  ident=cl.telefono_cliente or ident=cl.cedula) and
dbms_lob.compare(new_pass, cl.contraseńa) = 0;
      return c_id;
   END IF;
   return 0;
end;
/

-- LOGIN DEL PYME
CREATE OR REPLACE FUNCTION FC_PYME_LOGIN(ident VARCHAR2, pass VARCHAR2)
RETURN number is
resultado number;
p_id number;
new_pass BLOB;
begin
   new_pass:= TO_BLOB(utl_raw.cast_to_raw(pass));
   select count(*) into resultado from Pyme py where (ident=py.cedula_juridica or ident=py.telefono_pyme or ident=py.email)
and dbms_lob.compare(new_pass, py.clave) = 0;
   if resultado = 1 then
      select pyme_id into p_id from Pyme py where (ident=py.cedula_juridica or ident=py.telefono_pyme or ident=py.email)
and dbms_lob.compare(new_pass, py.clave) = 0;
      return p_id;
   END IF;
   return 0;
end;
/

commit;
```

```
CREATE OR REPLACE PROCEDURE SP_INSERT_PRODUCT (categoria VARCHAR2 ,l_name VARCHAR2, l_descripcion
VARCHAR2, precio number, pyme number) as
cat_id number;
aux number;
begin
   select count(*) into aux from Categoria ct where ct.nombre_categoria=categoria;
   if aux=0 then
      insert into Categoria(nombre_categoria) values (categoria);
      select ct.categoria_id into cat_id from Categoria ct where ct.nombre_categoria=categoria;
   else
      select ct.categoria_id into cat_id from Categoria ct where ct.nombre_categoria=categoria;
   end if;
   insert into Producto (nombre_producto, descripcion, precio_unitario, categoria_id, pyme_id) values(l_name, l_descripcion,
precio, cat_id, pyme);
end;
/

commit;
```

```
CREATE OR REPLACE FUNCTION SEARCH_PRODUCT(
descp varchar2
) return number
is
aux number;
pr_id number;
BEGIN
   select count(*) into aux from Producto pr where descp=pr.nombre_producto or pr.descripcion like
CONCAT('%',CONCAT(descp,'%'));
   if aux>1 then
      RETURN 0;
   end if;

   if aux=1 then
      select producto_id into pr_id from Producto pr where descp=pr.nombre_producto or pr.descripcion like
CONCAT('%',CONCAT(descp,'%'));
      return pr_id;
   end if;
   return 0;
END;
/

commit;
```

```
CREATE OR REPLACE PROCEDURE CREATE_ORDER(
id_cliente number
) as
fecha date;
begin
   insert into Orden( monto_total, estado, fecha_orden, cliente_id) values ( 0, 1, CURRENT_DATE, id_cliente);
end;
/

CREATE OR REPLACE PROCEDURE CREATE_LINE(
orden number,
nombre VARCHAR2,
cantidad NUMBER
) as
cash number(10,2);
product number;
begin
   SELECT SEARCH_PRODUCT(nombre) INTO product FROM dual;
   IF product!=0 THEN
      SELECT (pr.precio_unitario * cantidad) into cash from Producto pr where product=pr.producto_id;
   insert into Linea_de_producto(cantidad, monto, orden_id, producto_id) values (cantidad, cash, orden, product);
   END IF;
end;
/

CREATE or REPLACE TRIGGER TG_MONTO_ORDEN
   AFTER
   INSERT
   ON Linea_de_producto
   FOR EACH ROW
DECLARE
BEGIN
  update Orden ord set ord.monto_total = ord.monto_total + :new.monto where ord.orden_id=:new.orden_id;
END;
/
```

```
CREATE OR REPLACE PROCEDURE SP_CANCEL_ORDER(
order_id number
) as
begin
   UPDATE Orden od set estado=0 where order_id=od.orden_id;
end;
```

```
CREATE OR REPLACE PROCEDURE SP_EDIT_PRODUCT(
product number,
n_desc VARCHAR2,
n_mont NUMBER,
opc NUMBER
) as
old_mont NUMBER;
begin
   IF opc=1 then
      UPDATE Producto pr set nombre_producto=n_desc where product=pr.producto_id;
   END IF;

   IF opc=2 then
      UPDATE Producto pr set descripcion=n_desc where product=pr.producto_id;
   END IF;

   IF opc=3 then
      UPDATE Producto pr set pr.precio_unitario=n_mont where product=pr.producto_id;

   END IF;
end;
/

commit;
```

```
CREATE or REPLACE TRIGGER TG_CAMBIO_MONTO
   AFTER
   UPDATE OF precio_unitario
   ON Producto
   FOR EACH ROW
DECLARE
BEGIN
  insert into Registro( monto_anterior, fecha_registro, producto_id) values (:old.precio_unitario, CURRENT_DATE,
:old.producto_id);
END;
/


--Pruebas del trigger
--select * from Registro;
--select * from Producto where Producto.producto_id = 5;
--select * from Producto;

--exec producto_update_precio (5, 2500);

--UPDATE Producto
--SET
--    precio_unitario = 5500
--WHERE
--    producto_id = 5;

commit;
```

```
create or replace procedure SP_VIEW_SALES (pyme number, mes number) as
c1 SYS_REFCURSOR;
articulos_vendidos number;
pmonto_total number;
name_prod VARCHAR2(55);
mesesito number;
begin

   OPEN c1 for
   select pr.nombre_producto Porducto, SUM(cantidad) Cantidad, SUM(pr.precio_unitario*cantidad) MontoTotal, mes Mes
into name_prod, articulos_vendidos, pmonto_total, mesesito from linea_de_producto ldp
   inner join Producto pr on ldp.producto_id=pr.producto_id
   inner join Orden ord on ldp.orden_id=ord.orden_id
   where pr.pyme_id=pyme and mes=EXTRACT(MONTH FROM ord.fecha_orden) and ord.estado=1
   group by pr.nombre_producto;

   DBMS_SQL.RETURN_RESULT(c1);

end;
/
```

```
create or replace procedure SP_SELECT_ORDERS (cliente number, condition number) as
c1 SYS_REFCURSOR;
begin

   if condition=0 then

      open c1 for
      SELECT CONCAT('Orden#',ord.orden_id) Orden, ord.monto_total Monto, ord.fecha_orden Fecha, estado FROM Orden
ord where ord.cliente_id=cliente;
      DBMS_SQL.RETURN_RESULT(c1);

   END IF;

   if condition=1 then

      open c1 for
      SELECT CONCAT('Orden#',ord.orden_id) Orden, ord.monto_total Monto, ord.fecha_orden Fecha, estado FROM Orden
ord where ord.cliente_id=cliente and ord.estado=0;
      DBMS_SQL.RETURN_RESULT(c1);

   END IF;


   if condition=2 then

      open c1 for
      SELECT CONCAT('Orden#',ord.orden_id) Orden, ord.monto_total Monto, ord.fecha_orden Fecha, estado FROM Orden
ord where ord.cliente_id=cliente and ord.estado=1;
      DBMS_SQL.RETURN_RESULT(c1);

   END IF;

end;
/
```

```
CREATE OR REPLACE PACKAGE paquete_modificacion AS
   PROCEDURE sp_insert_product (
      categoria     VARCHAR2,
      l_name        VARCHAR2,
      l_descripcion VARCHAR2,
      precio        NUMBER,
      pyme          NUMBER
   );

   PROCEDURE create_order (
      id_cliente NUMBER
   );

   PROCEDURE create_line (
      orden    NUMBER,
      nombre   VARCHAR2,
      cantidad NUMBER
   );

   PROCEDURE sp_cancel_order (
      order_id NUMBER
   );

   PROCEDURE sp_edit_product (
      product NUMBER,
      n_desc  VARCHAR2,
      n_mont  NUMBER,
      opc     NUMBER
   );

END paquete_modificacion;
/

CREATE OR REPLACE PACKAGE BODY paquete_modificacion AS

   PROCEDURE sp_insert_product (
      categoria     VARCHAR2,
      l_name        VARCHAR2,
      l_descripcion VARCHAR2,
      precio        NUMBER,
      pyme          NUMBER
   ) AS
      cat_id NUMBER;
      aux    NUMBER;
   BEGIN
      SELECT
         COUNT(*)
      INTO aux
      FROM
         categoria ct
      WHERE
         ct.nombre_categoria = categoria;

      IF aux = 0 THEN
         INSERT INTO categoria ( nombre_categoria ) VALUES ( categoria );

         SELECT
            ct.categoria_id
         INTO cat_id
         FROM
            categoria ct
         WHERE
            ct.nombre_categoria = categoria;

      ELSE
         SELECT
```

```
            ct.categoria_id
        INTO cat_id
        FROM
            categoria ct
        WHERE
            ct.nombre_categoria = categoria;

    END IF;

    INSERT INTO producto (
        nombre_producto,
        descripcion,
        precio_unitario,
        categoria_id,
        pyme_id
    ) VALUES (
        l_name,
        l_descripcion,
        precio,
        cat_id,
        pyme
    );

END sp_insert_product;

PROCEDURE create_order (
    id_cliente NUMBER
) AS
    fecha DATE;
BEGIN
    INSERT INTO orden (
        monto_total,
        estado,
        fecha_orden,
        cliente_id
    ) VALUES (
        0,
        1,
        current_date,
        id_cliente
    );

END create_order;

PROCEDURE create_line (
    orden    NUMBER,
    nombre   VARCHAR2,
    cantidad NUMBER
) AS
    cash    NUMBER(10, 2);
    product NUMBER;
BEGIN
    SELECT
        search_product(nombre)
    INTO product
    FROM
        dual;

    IF product != 0 THEN
        SELECT
            ( pr.precio_unitario * cantidad )
        INTO cash
        FROM
            producto pr
        WHERE
```

```
            product = pr.producto_id;

        INSERT INTO linea_de_producto (
            cantidad,
            monto,
            orden_id,
            producto_id
        ) VALUES (
            cantidad,
            cash,
            orden,
            product
        );

    END IF;

END create_line;

PROCEDURE sp_cancel_order (
    order_id NUMBER
) AS
BEGIN
    UPDATE orden od
    SET
        estado = 0
    WHERE
        order_id = od.orden_id;

END sp_cancel_order;

PROCEDURE sp_edit_product (
    product NUMBER,
    n_desc  VARCHAR2,
    n_mont  NUMBER,
    opc     NUMBER
) AS
    old_mont NUMBER;
BEGIN
    IF opc = 1 THEN
        UPDATE producto pr
        SET
            nombre_producto = n_desc
        WHERE
            product = pr.producto_id;

    END IF;

    IF opc = 2 THEN
        UPDATE producto pr
        SET
            descripcion = n_desc
        WHERE
            product = pr.producto_id;

    END IF;

    IF opc = 3 THEN
        UPDATE producto pr
        SET
            pr.precio_unitario = n_mont
        WHERE
            product = pr.producto_id;

    END IF;
```

```
    END sp_edit_product;

END paquete_modificacion;
/

CREATE OR REPLACE PACKAGE paquete_consulta AS
    FUNCTION search_product (
        descp VARCHAR2
    ) RETURN NUMBER;

    FUNCTION fc_valid_cliente (
        new_cedula          VARCHAR2,
        new_apodo           VARCHAR2,
        new_telefono_cliente VARCHAR2
    ) RETURN NUMBER;

    FUNCTION fc_valid_pyme (
        new_cedula          VARCHAR2,
        new_email           VARCHAR2,
        new_telefono_cliente VARCHAR2
    ) RETURN NUMBER;

    FUNCTION fc_cliente_login (
        ident VARCHAR2,
        pass  VARCHAR2
    ) RETURN NUMBER;

    FUNCTION fc_pyme_login (
        ident VARCHAR2,
        pass  VARCHAR2
    ) RETURN NUMBER;

    PROCEDURE sp_select_orders (
        cliente   NUMBER,
        condition NUMBER
    );

    PROCEDURE sp_view_sales (
        pyme NUMBER,
        mes  NUMBER
    );

END paquete_consulta;
/

CREATE OR REPLACE PACKAGE BODY paquete_consulta AS

    FUNCTION search_product (
        descp VARCHAR2
    ) RETURN NUMBER IS
        aux   NUMBER;
        pr_id NUMBER;
    BEGIN
        SELECT
            COUNT(*)
        INTO aux
        FROM
            producto pr
        WHERE
            descp = pr.nombre_producto
            OR pr.descripcion LIKE concat('%', concat(descp, '%'));

        IF aux > 1 THEN
            RETURN 0;
        END IF;
```

```
    IF aux = 1 THEN
        SELECT
            producto_id
        INTO pr_id
        FROM
            producto pr
        WHERE
            descp = pr.nombre_producto
            OR pr.descripcion LIKE concat('%', concat(descp, '%'));

        RETURN pr_id;
    END IF;

    RETURN 0;
END search_product;

FUNCTION fc_valid_cliente (
    new_cedula          VARCHAR2,
    new_apodo           VARCHAR2,
    new_telefono_cliente VARCHAR2
) RETURN NUMBER IS
    apodex  NUMBER;
    cedulex NUMBER;
    numberx NUMBER;
BEGIN
    SELECT
        COUNT(*)
    INTO apodex
    FROM
        cliente
    WHERE
        new_apodo = apodo;

    SELECT
        COUNT(*)
    INTO cedulex
    FROM
        cliente
    WHERE
        new_cedula = cedula;

    SELECT
        COUNT(*)
    INTO numberx
    FROM
        cliente
    WHERE
        new_telefono_cliente = telefono_cliente;

    IF
        apodex = 0
        AND cedulex = 0
        AND numberx = 0
    THEN
        RETURN 0;
    ELSE
        IF apodex != 0 THEN
            RETURN 1;
        END IF;
        IF cedulex != 0 THEN
            RETURN 2;
        END IF;
        IF numberx != 0 THEN
            RETURN 3;
        END IF;
```

```
    END IF;

END fc_valid_cliente;

FUNCTION fc_valid_pyme (
    new_cedula          VARCHAR2,
    new_email           VARCHAR2,
    new_telefono_cliente VARCHAR2
) RETURN NUMBER IS
    emailex NUMBER;
    cedulex NUMBER;
    numberx NUMBER;
BEGIN
    SELECT
        COUNT(*)
    INTO emailex
    FROM
        pyme
    WHERE
        new_email = email;

    SELECT
        COUNT(*)
    INTO cedulex
    FROM
        pyme
    WHERE
        new_cedula = cedula_juridica;

    SELECT
        COUNT(*)
    INTO numberx
    FROM
        pyme
    WHERE
        new_telefono_cliente = telefono_pyme;

    IF
        emailex = 0
        AND cedulex = 0
        AND numberx = 0
    THEN
        RETURN 0;
    ELSE
        IF emailex != 0 THEN
            RETURN 1;
        END IF;
        IF cedulex != 0 THEN
            RETURN 2;
        END IF;
        IF numberx != 0 THEN
            RETURN 3;
        END IF;
    END IF;

END fc_valid_pyme;

FUNCTION fc_cliente_login (
    ident VARCHAR2,
    pass  VARCHAR2
) RETURN NUMBER IS
    resultado NUMBER;
    c_id      NUMBER;
    new_pass  BLOB;
BEGIN
```

```
    new_pass := to_blob(utl_raw.cast_to_raw(pass));
    SELECT
        COUNT(*)
    INTO resultado
    FROM
        cliente cl
    WHERE
        ( ident = cl.apodo
          OR ident = cl.telefono_cliente
          OR ident = cl.cedula )
        AND dbms_lob.compare(new_pass, cl.contraseña) = 0;

    IF resultado = 1 THEN
        SELECT
            cliente_id
        INTO c_id
        FROM
            cliente cl
        WHERE
            ( ident = cl.apodo
              OR ident = cl.telefono_cliente
              OR ident = cl.cedula )
            AND dbms_lob.compare(new_pass, cl.contraseña) = 0;

        RETURN c_id;
    END IF;

    RETURN 0;
END fc_cliente_login;

FUNCTION fc_pyme_login (
    ident VARCHAR2,
    pass  VARCHAR2
) RETURN NUMBER IS
    resultado NUMBER;
    p_id      NUMBER;
    new_pass  BLOB;
BEGIN
    new_pass := to_blob(utl_raw.cast_to_raw(pass));
    SELECT
        COUNT(*)
    INTO resultado
    FROM
        pyme py
    WHERE
        ( ident = py.cedula_juridica
          OR ident = py.telefono_pyme
          OR ident = py.email )
        AND dbms_lob.compare(new_pass, py.clave) = 0;

    IF resultado = 1 THEN
        SELECT
            pyme_id
        INTO p_id
        FROM
            pyme py
        WHERE
            ( ident = py.cedula_juridica
              OR ident = py.telefono_pyme
              OR ident = py.email )
            AND dbms_lob.compare(new_pass, py.clave) = 0;

        RETURN p_id;
    END IF;
```

```
    RETURN 0;
END fc_pyme_login;

PROCEDURE sp_select_orders (
    cliente   NUMBER,
    condition NUMBER
) AS
    c1 SYS_REFCURSOR;
BEGIN
    IF condition = 0 THEN
        OPEN c1 FOR SELECT
                      concat('Orden#', ord.orden_id) orden,
                      ord.monto_total              monto,
                      ord.fecha_orden              fecha,
                      estado
                  FROM
                      orden ord
                  WHERE
                      ord.cliente_id = cliente;

        dbms_sql.return_result(c1);
    END IF;

    IF condition = 1 THEN
        OPEN c1 FOR SELECT
                      concat('Orden#', ord.orden_id) orden,
                      ord.monto_total              monto,
                      ord.fecha_orden              fecha,
                      estado
                  FROM
                      orden ord
                  WHERE
                          ord.cliente_id = cliente
                      AND ord.estado = 0;

        dbms_sql.return_result(c1);
    END IF;

    IF condition = 2 THEN
        OPEN c1 FOR SELECT
                      concat('Orden#', ord.orden_id) orden,
                      ord.monto_total              monto,
                      ord.fecha_orden              fecha,
                      estado
                  FROM
                      orden ord
                  WHERE
                          ord.cliente_id = cliente
                      AND ord.estado = 1;

        dbms_sql.return_result(c1);
    END IF;

END sp_select_orders;

PROCEDURE sp_view_sales (
    pyme NUMBER,
    mes  NUMBER
) AS
    c1               SYS_REFCURSOR;
    articulos_vendidos NUMBER;
    pmonto_total       NUMBER;
    name_prod          VARCHAR2(55);
    mesesito           NUMBER;
```

```
  BEGIN
    OPEN c1 FOR SELECT
               pr.nombre_producto              porducto,
               SUM(cantidad)                   cantidad,
               SUM(pr.precio_unitario * cantidad) montototal,
               mes                             mes
            INTO
               name_prod,
               articulos_vendidos,
               pmonto_total,
               mesesito
            FROM
                linea_de_producto ldp
               INNER JOIN producto pr ON ldp.producto_id = pr.producto_id
               INNER JOIN orden    ord ON ldp.orden_id = ord.orden_id
            WHERE
                pr.pyme_id = pyme
               AND mes = EXTRACT(MONTH FROM ord.fecha_orden)
               AND ord.estado = 1
            GROUP BY
               pr.nombre_producto;

    dbms_sql.return_result(c1);
  END sp_view_sales;

END paquete_consulta;
/
```