

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

Bases de Datos

4ta Práctica Dirigida

(Segundo Semestre 2021)

Indicaciones generales:

- Duración: **50 minutos** (parte dirigida).
- Se utilizará el mismo caso del laboratorio dirigido anterior para enfocarnos en el tema de hoy: procedimientos almacenados.
- Antes de comenzar el laboratorio, ejecute Oracle SQL Developer, cree una nueva conexión llamada **LAB4_DIRIGIDA**, y ejecute el script contenido en el archivo **INF246_2021-2_LAB4_DIRIGIDA_DDL.sql** y luego **INF246_2021-2_LAB4_DIRIGIDA_DML.sql**.

Sistema de Registro de Pedidos

Una empresa de distribución de productos cuenta con el siguiente sistema de información web y móvil para la toma de pedidos que realizan sus clientes. Las funcionalidades principales son:

- ✓ Registro y consulta de clientes (y sus respectivos datos de contacto incluyendo sus direcciones).
- ✓ Registro y consulta de productos (y sus datos de almacenamiento, stocks y precios).
- ✓ Registro y seguimiento de los pedidos que se realizan a través de los sistemas (productos, dirección de entrega, montos, entre otros).

En la Figura1, se presenta el modelo relacional del sistema.

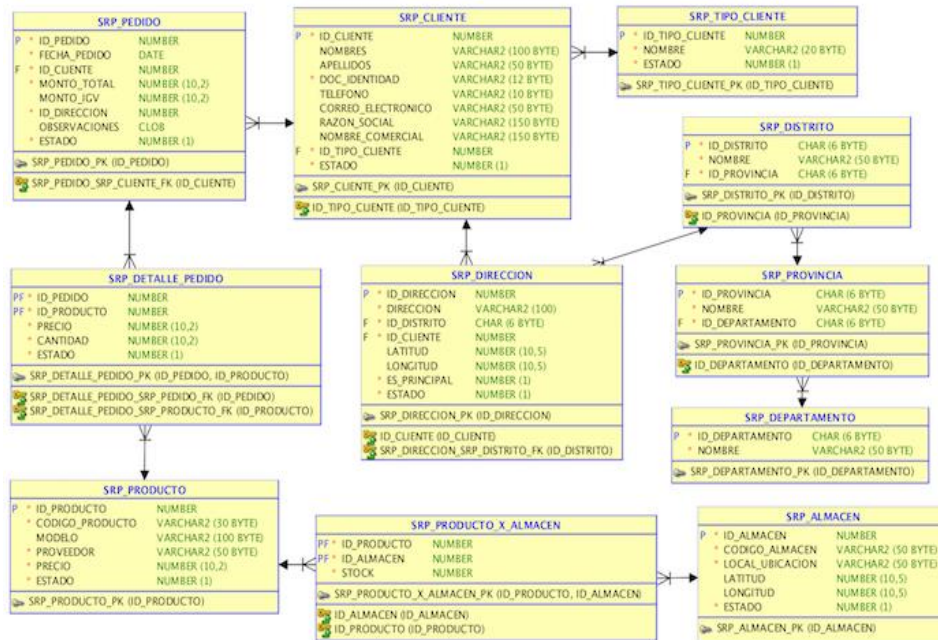


Figura 1. Modelo relacional del sistema

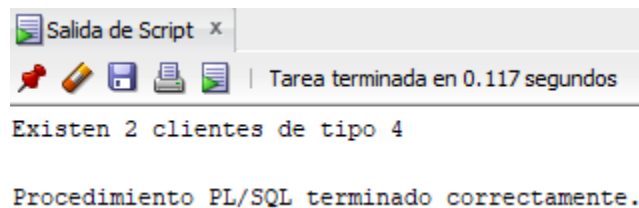
Ejercicio 1

Elaborar una función que reciba como parámetro un tipo de cliente (valores 1, 2, 3, o 4) y que devuelva la cantidad de clientes con dicho tipo. Si el valor del parámetro no es correcto la función deberá retornar -1.

```
create or replace function f_contar_clientes_por_tipo(n_id_tipo_cliente number)
return number
as
    n_cantidad number;
begin
    if n_id_tipo_cliente < 1 or n_id_tipo_cliente > 4 then
        return -1;
    else
        select count(*) into n_cantidad
        from SRP_Cliente
        where id_tipo_cliente = n_id_tipo_cliente;
        return n_cantidad;
    end if;
end;
```

Para probar la función utilice el siguiente código, puede cambiar el valor de la variable n_id_tipo_cliente:

```
set SERVEROUTPUT ON
declare
    n_cantidad number;
    n_id_tipo_cliente number;
begin
    n_id_tipo_cliente := 4;
    n_cantidad := f_contar_clientes_por_tipo_cliente(n_id_tipo_cliente);
    if n_cantidad <> -1 then
        dbms_output.put_line( 'Existen ' || n_cantidad || ' clientes de tipo ' || n_id_tipo_cliente);
    else
        dbms_output.put_line( 'ID Tipo Cliente no válido' );
    end if;
end;
```



Ejercicio 2

Elaborar una función que reciba como parámetro una fecha y que devuelva dicha fecha en forma de frase. Por ejemplo, si la fecha que recibe la función es '01/10/2021', esta devolverá la frase '01 de octubre de 2021'.

```
create or replace function f_fecha_frase( v_fecha date )
return varchar2
is
    v_d number;
    v_m number;
```

```

v_a number;
v_dia varchar2(2);
v_mes varchar2(20);
v_año varchar2(4);
v_frase varchar2(30);
begin
v_d := extract( day from v_fecha );
v_m := extract( month from v_fecha );
v_a := extract( year from v_fecha );
case v_m
    when 1 then v_mes := 'enero';
    when 2 then v_mes := 'febrero';
    when 3 then v_mes := 'marzo';
    when 4 then v_mes := 'abril';
    when 5 then v_mes := 'mayo';
    when 6 then v_mes := 'junio';
    when 7 then v_mes := 'julio';
    when 8 then v_mes := 'agosto';
    when 9 then v_mes := 'setiembre';
    when 10 then v_mes := 'octubre';
    when 11 then v_mes := 'noviembre';
    else v_mes := 'diciembre';
end case;
v_dia := to_char( v_d );
v_año := to_char( v_a );
v_frase := v_dia || ' de ' || v_mes || ' de ' || v_año;
return v_frase;
end;





```

Para probar la función utilice el siguiente código:

```

select
    id_pedido,
    f_fecha_frase(fecha_pedido) as Fecha_Pedido
from SRP_Pedido;

```

Resultado de la Consulta x	
    SQL Todas las Filas Recuperadas: 1	
ID_PEDIDO	FECHA_PEDIDO
1	1 5 de abril de 2021
2	2 6 de abril de 2021
3	3 6 de mayo de 2021
4	4 6 de mayo de 2021
5	5 7 de junio de 2021
6	6 7 de junio de 2021
7	7 7 de julio de 2021
8	8 7 de julio de 2021
9	9 7 de agosto de 2021
10	10 8 de setiembre de 2021
11	11 4 de octubre de 2021
12	12 4 de octubre de 2021
13	13 1 de octubre de 2021
14	14 1 de octubre de 2021
15	15 1 de octubre de 2021

Ejercicio 3

Elaborar un procedimiento que reciba como parámetro el nombre de un distrito y que muestre en pantalla la cantidad de personas que tiene direcciones en dicho distrito.

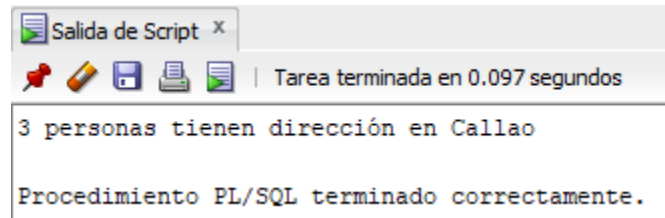
```
create or replace procedure p_cantidad_personas_distrito(v_nom_distrito varchar2)
is
    v_cantidad number;
begin
    select count(*) into v_cantidad
    from
        SRP_Direccion Dir,
        SRP_Distrito Dis
    where
        Dir.id_distrito = Dis.id_distrito
        and Dis.nombre = v_nom_distrito;

    dbms_output.put_line(v_cantidad || ' personas tienen dirección en ' || v_nom_distrito );
end;
```

Para probar la función utilice el siguiente código, puede cambiar el valor del parámetro por 'Lima':

```
set SERVEROUTPUT ON
```

```
exec p_cantidad_personas_distrito( 'Callao' );
```



Ejercicio 4

Elaborar un procedimiento que reciba como parámetro el id de pedido y muestre la provincia donde se entregará dicho pedido. Si el id pedido no existe mostrar un mensaje apropiado.

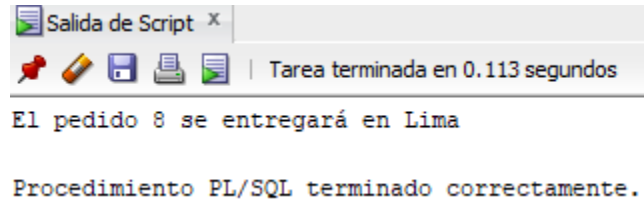
```
create or replace procedure p_mostrar_provincia_pedido(v_id_pedido number)
is
    v_nom_prov varchar2(50);
begin
    select Pro.nombre into v_nom_prov
    from
        SRP_Pedido P, SRP_Direccion D, SRP_Distrito Dis, SRP_Provincia Pro
    where
        P.id_pedido = v_id_pedido
        and P.id_cliente = D.id_direccion
        and D.id_distrito = Dis.id_distrito
        and Dis.id_provincia = Pro.id_provincia;

    dbms_output.put_line('El pedido ' || v_id_pedido || ' se entregará en ' || v_nom_prov);
exception
    when NO_DATA_FOUND then
        dbms_output.put_line( 'El pedido no existe' );
end;
```

Para probar la función utilice el siguiente código, puede cambiar el valor del parámetro id_pedido

```
set SERVEROUTPUT ON
```

```
exec p_mostrar_provincia_pedido(8);
```



Ejercicio 5

Elaborar un procedimiento que reciba como parámetros el id de un almacén, el id de un producto y una cantidad de producto. En base a ello se debe agregar dicha cantidad al stock del producto de determinado almacén. Si no se encuentra el producto ni el almacén se debe insertar en la tabla con dicho stock.

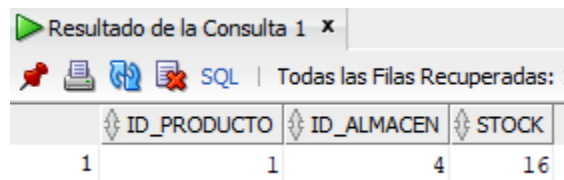
```
create or replace procedure p_incrementar_stock(v_id_almacen number, v_id_producto number,
v_stock number)
is
    v_encontrado number;
begin
    select count(*) into v_encontrado
    from SRP_PRODUCTO_X_ALMACEN
    where id_almacen = v_id_almacen
          and id_producto = v_id_producto;

    update SRP_PRODUCTO_X_ALMACEN
    set stock = stock + v_stock
    where id_almacen = v_id_almacen
          and id_producto = v_id_producto;
exception
    when NO_DATA_FOUND then
```

```
insert into SRP_PRODUCTO_X_ALMACEN (id_producto, id_almacen, stock)
values (v_id_producto, v_id_almacen, v_stock);
end;
```

Para probar el procedimiento primero validamos el stock del producto 1 en el almacén 4:

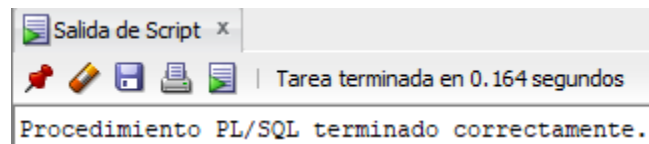
```
select * from SRP_PRODUCTO_X_ALMACEN where id_producto = 1 and id_almacen = 4
```



ID_PRODUCTO	ID_ALMACEN	STOCK
1	4	16

Luego ejecutamos el procedimiento para aumentar el stock en 2:

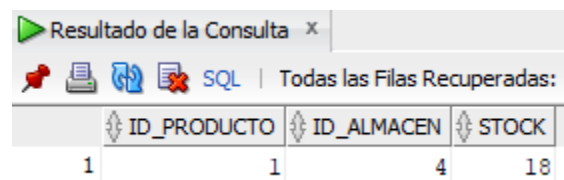
```
set SERVEROUTPUT ON
exec p_incrementar_stock(4, 1, 2);
```



Procedimiento PL/SQL terminado correctamente.

Volvemos a validar el stock y visualizamos que aumentó en 2:

```
select * from SRP_PRODUCTO_X_ALMACEN where id_producto = 1 and id_almacen = 4
```



ID_PRODUCTO	ID_ALMACEN	STOCK
1	4	18

Ejercicio 6

Elaborar un procedimiento que devuelva dos valores: el pedido con el mayor monto total y el id del cliente que lo realizó.

```
create or replace procedure p_obtener_mayor_pedido(
    v_mayor_monto_pedido out number,
    v_id_cliente out number)
is
begin
    select max(monto_total) into v_mayor_monto_pedido
    from SRP_Pedido;

    select C.id_cliente into v_id_cliente
    from SRP_Pedido P, SRP_Cliente C
    where P.monto_total = v_mayor_monto_pedido
    and P.id_cliente = C.id_cliente;
end;
```

Para probar el procedimiento utiliza el siguiente código:

```
set SERVEROUTPUT ON
declare
    v_mayor_monto_pedido number;
    v_id_cliente number;
begin
    p_obtener_mayor_pedido(v_mayor_monto_pedido, v_id_cliente);
    dbms_output.put_line( 'El pedido con mayor monto es de ' || v_mayor_monto_pedido || '
y fue realizado por el id cliente ' || v_id_cliente);
end;
```

