

```

--1)
select p.product_id, p.description, pi.list_price, pi.min_price, count(s.product_id) Cantidad
from product p inner join price pi on (p.product_id = pi.product_id)
        inner join item s on (s.product_id = pi.product_id)
where pi.end_date is null
group by p.product_id, p.description, pi.list_price, pi.min_price
having count(s.product_id) < 10
order by description

--2)
create or replace function validate_customer1
(pi_cust_name in customer.name%type)
return customer.name%type
is
    v_cust_id customer.customer_id%type;
begin

    select customer_id
    into v_cust_id
    from customer
    where lower(name) = lower(pi_cust_name);

    return v_cust_id;

exception
    when too_many_rows then raise_application_error(-20001, 'Hay mas de un cliente para ese ese nombre');
    when no_data_found then raise_application_error(-20002, 'No existe el cliente con el nombre indicado');
    when others then raise_application_error(-20003, 'Paso algo');

end;

begin
    dbms_output.put_line(validate_customer1('VOLLYRITE'));

end;

--3)

create or replace function validate_customer2
(pi_cust_name in customer.name%type)
return customer.name%type
is
    v_cust_id customer.customer_id%type;
begin

    select customer_id
    into v_cust_id
    from customer
    where lower(name) = lower(pi_cust_name);

    return v_cust_id;

exception
    when too_many_rows then return 0;
    when no_data_found then return null;

end;

create or replace procedure create_customer
(pi_id_cust customer.customer_id%type, pi_name customer.name%type, pi_sp_id employee.employee_id%type)
is
    name_already_exists exception;
    pragma exception_init(name_already_exists, -20001);

    no_pk_ex exception;
    pragma exception_init(no_pk_ex, -02291);

    e_unique_pk exception;
    pragma exception_init(e_unique_pk, -00001);

    v_valid_name customer.customer_id%type;
begin

    v_valid_name := validate_customer2(pi_name);

    if v_valid_name is not null or v_valid_name = 0 then
        raise_application_error(-20001, 'Error');
    end if;

    insert into customer(customer_id, name, salesperson_id)
    values (pi_id_cust, pi_name, pi_sp_id);

    if(sql%rowcount = 1) then
        dbms_output.put_line('Se creo el cliente correctamente');
    end if;

exception
    when name_already_exists then dbms_output.put_line('Ya existe un cliente para el nombre indicado');

```

```

        when no_pk_ex then dbms_output.put_line('No existe el vendedor indicado');
        when e_unique_pk then dbms_output.put_line('Ya existe un cliente con el id indicado');
        when others then dbms_output.put_line('Ocurrio un error inesperado');
end;

begin

create_customer(45, 'aaa', 7505);
end;

--4)
declare

    v_prod_id product.product_id%type;

    cursor c_prod (pi_prod_id product.product_id%type)
    is
    select *
    from price
    where product_id = v_prod_id
    order by end_date asc;

    v_x varchar(1);
begin

    v_prod_id := :Producto;

    select '*'
    into v_x
    from product
    where product_id = v_prod_id;

    dbms_output.put_line('El producto: ' || v_prod_id || ' tuvo este historial: ');
    for r_prod in c_prod(v_prod_id) loop
        dbms_output.put_line('Precio de lista: ' || r_prod.list_price || ', Precio minimo: ' || r_prod.min_price || ', Vigencia ' || r_prod.
start_date || ' - ' || nvl(to_char(r_prod.end_date), 'Sin tiene fecha de fin'));
    end loop;

    exception
        when no_data_found then dbms_output.put_line('No existe el producto indicado');
        when others then dbms_output.put_line('Error inesperado');
end;

```