Si no la tienes ya, carga en SQL Developer el script de la base de datos **jardineria\_oracle.sql**.

Realiza las acciones que se piden a continuación y entrega la actividad rellena, incluyendo el código, explicaciones y capturas de pantalla.

El siguiente código pretende validar que todos los clientes que tienen pagos, también tienen pedidos:

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  CURSOR CurPago IS  SELECT CodigoCliente, cantidad FROM PAGOS;  Codigo pedidos.CodigoPedido%TYPE;  BEGIN  FOR Registro IN CurPago LOOP  SELECT CodigoPedido INTO Codigo FROM pedidos  WHERE CodigoCliente = Registro.CodigoCliente;    DBMS\_OUTPUT.PUT\_LINE(Registro.CodigoCliente||' '||Registro.cantidad||' '||Codigo);  END LOOP;    EXCEPTION  WHEN OTHERS THEN  dbms\_output.put\_line('Código de error: '||SQLCODE);  dbms\_output.put\_line('Mensaje de error: '||SQLERRM);  END; |

¿Qué error devuelve y por qué se produce? ¿Cómo se podría solucionar? Arregla el código para que sirva para comprobar si todos los clientes que tienen pagos, tienen pedidos.

|  |
| --- |
| Este código tenía varios errores, el primero era que faltaba poner “clientes.CodigoCliente”  Ya que sino se ponía nos decía que el campo era ambiguo, segundo error es que no se había hecho select de CodigoPedido, ya que mas tarde lo declaramos como %TYPE, el tercer error era que no se había hecho joinn de los campos pedidos con codigocliente y pagos con codigocliente y el ultimo error es que ahora de imprimir los datos por consola en el for que lo hemos llamado como Registro no se había puesto “Registro.Codigocliente”  DECLARE  CURSOR CurPago IS  SELECT clientes.CodigoCliente, cantidad, CodigoPedido FROM clientes  inner join pedidos on clientes.codigocliente = pedidos.codigocliente  inner join pagos on clientes.codigocliente = pagos.codigocliente;  Codigo pedidos.CodigoPedido%TYPE;  BEGIN  FOR Registro IN CurPago LOOP  DBMS\_OUTPUT.PUT\_LINE(Registro.CodigoCliente||' '||Registro.cantidad||' '||Registro.CodigoCliente);  END LOOP;  EXCEPTION  WHEN OTHERS THEN  dbms\_output.put\_line('Código de error: '||SQLCODE);  dbms\_output.put\_line('Mensaje de error: '||SQLERRM);  END;  / |

Modifica el código inicial para mantener el funcionamiento tal cual está (seguirá produciendo error), pero controla la excepción que se produce explícitamente y muestra por pantalla el texto ‘Error controlado’.

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  CURSOR CurPago IS  SELECT CodigoCliente, cantidad FROM PAGOS;  Codigo pedidos.CodigoPedido%TYPE;  BEGIN  FOR Registro IN CurPago LOOP  SELECT CodigoPedido INTO Codigo FROM pedidos  WHERE CodigoCliente = Registro.CodigoCliente;    DBMS\_OUTPUT.PUT\_LINE(Registro.CodigoCliente||' '||Registro.cantidad||' '||Codigo);  END LOOP;    EXCEPTION  when too\_many\_rows  then dbms\_output.put\_line('Error controlado');  WHEN OTHERS THEN  dbms\_output.put\_line('Código de error: '||SQLCODE);  dbms\_output.put\_line('Mensaje de error: '||SQLERRM);  END;  / |

Si no la tienes ya, carga en SQL Developer el script de la base de datos **jardineria\_oracle.sql**.

Realiza un programa que recorra la tabla de clientes y muestre por pantalla su código, nombre y un texto que indique:

- ‘Nacional’: Si su país es USA.

- ‘Internacional’: Si su país no es USA.

Por ejemplo:

1 DGPRODUCTIONS GARDEN Nacional

3 Gardening Associates Nacional

4 Gerudo Valley Nacional

5 Tendo Garden Nacional

6 Lasas S.A. Internacional

7 Beragua Internacional

…

|  |
| --- |
| Declare  CURSOR curclientes is  select nombrecliente,codigocliente,pais  from clientes;  begin  for c in curclientes loop  if c.pais='USA' then  dbms\_output.put\_line(c.nombrecliente||' '||'Nacional');  else  dbms\_output.put\_line(c.nombrecliente||' '||'Internacional');  end if;  end loop;  end; |

Si no la tienes ya, carga en SQL Developer el script de la base de datos **jardineria\_oracle.sql**.

Realizar un programa que recorra los empleados y muestre por pantalla su código y un texto en función de:

- Si tiene oficina en Madrid: ‘Madrid’.

- Si NO tiene oficina en Madrid: ‘Otro’.

Por ejemplo (el orden de los registros es indiferente):

4 Otro

5 Otro

6 Otro

7 Madrid

8 Madrid

…

|  |
| --- |
| Declare  CURSOR curempleados is  select \*  from empleados inner join oficinas  on empleados.codigooficina = oficinas.codigooficina;  begin  for e in curempleados loop  if e.ciudad='Madrid' then  dbms\_output.put\_line('Madrid');  else  dbms\_output.put\_line('otro');  end if;  end loop;  end; |

Si no la tienes ya, carga en SQL Developer el script de la base de datos **jardineria\_oracle.sql**.

Crea un procedimiento llamado muestrapedido, que le pasemos el código de un pedido, y nos muestre los datos más relevantes del cliente (Nombre, dirección, ciudad y país) del pedido, y además la fecha del pedido.

A continuación mostrará todos los detalles de los artículos facturados en ese pedido (cantidad, producto, precio y total de cada artículo), indicando al final el precio total de todos los artículos, el IVA y el precio más IVA.

Controla las excepciones que creas necesarias.

|  |
| --- |
| Declare  CURSOR muestrapedido is select clientes.nombrecliente,clientes.lineadireccion1,clientes.lineadireccion2,clientes.ciudad,clientes.pais,pedidos.fechapedido,detallepedidos.cantidad,productos.nombre,productos.precioventa,productos.cantidadenstock from detallepedidos  INNER JOIN pedidos  ON detallepedidos.codigopedido = pedidos.codigopedido  INNER JOIN productos  ON detallepedidos.codigoproducto = productos.codigoproducto  inner join clientes  on pedidos.codigocliente = clientes.codigocliente  where pedidos.codigopedido= '17';  begin  for datos in muestrapedido loop  dbms\_output.put\_line(datos.nombrecliente||' '||datos.lineadireccion1||' '||datos.lineadireccion2||' '||datos.ciudad||' '||datos.pais||' '||datos.fechapedido||' '||datos.cantidad||' '||datos.nombre||' '||datos.precioventa||' '||datos.cantidadenstock);  end loop;  end; |

Ejecuta el procedimiento anterior con el pedido 2 y muestra el resultado.

|  |
| --- |
| Declare  CURSOR muestrapedido is select clientes.nombrecliente,clientes.lineadireccion1,clientes.lineadireccion2,clientes.ciudad,clientes.pais,pedidos.fechapedido,detallepedidos.cantidad,productos.nombre,productos.precioventa,productos.cantidadenstock from detallepedidos  INNER JOIN pedidos  ON detallepedidos.codigopedido = pedidos.codigopedido  INNER JOIN productos  ON detallepedidos.codigoproducto = productos.codigoproducto  inner join clientes  on pedidos.codigocliente = clientes.codigocliente  where pedidos.codigopedido= '2';  begin  for datos in muestrapedido loop  dbms\_output.put\_line(datos.nombrecliente||' '||datos.lineadireccion1||' '||datos.lineadireccion2||' '||datos.ciudad||' '||datos.pais||' '||datos.fechapedido||' '||datos.cantidad||' '||datos.nombre||' '||datos.precioventa||' '||datos.cantidadenstock);  end loop;  end; |

Si no la tienes ya, carga en SQL Developer el script de la base de datos **jardineria\_oracle.sql**.

Realiza un procedimiento al que se le pase el código de un cliente y liste los datos del cliente: Código, Nombre, Ciudad y País.

A continuación mostrará los pagos que ha realizado, ordenados cronológicamente (IDTRANSACCION, FECHA, FORMA y CANTIDAD).

Para finalizar debe mostrar la cantidad total pagada.

Por ejemplo:



Implementa también el tratamiento de excepciones.

|  |
| --- |
| Declare  CURSOR muestraclientes is select clientes.codigocliente,clientes.nombrecliente,clientes.ciudad,clientes.pais,pagos.idtransaccion,pagos.fechapago,pagos.formapago,pagos.cantidad from clientes  INNER JOIN pagos  ON clientes.codigocliente = pagos.codigocliente  where pagos.codigocliente= '3';  begin  for datos in muestraclientes loop  dbms\_output.put\_line(datos.codigocliente||' '||datos.nombrecliente||' '||datos.ciudad||' '||datos.pais||' '||datos.idtransaccion||' '||datos.fechapago||' '||datos.formapago||' '||datos.cantidad);  end loop;  dbms\_output.put\_line('Total pagos efectuados:'||' '|| (5000+5000+926));  end; |

Ejecútalo para el cliente 3 y muestra el resultado.

|  |
| --- |
|  |