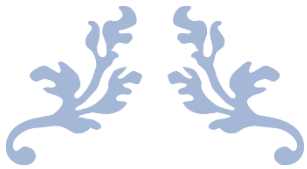


TAREA AD05_EJERCICIO2

Módulo de acceso a datos en modalidad a distancia del I.E.S. Augusto González de Linares.

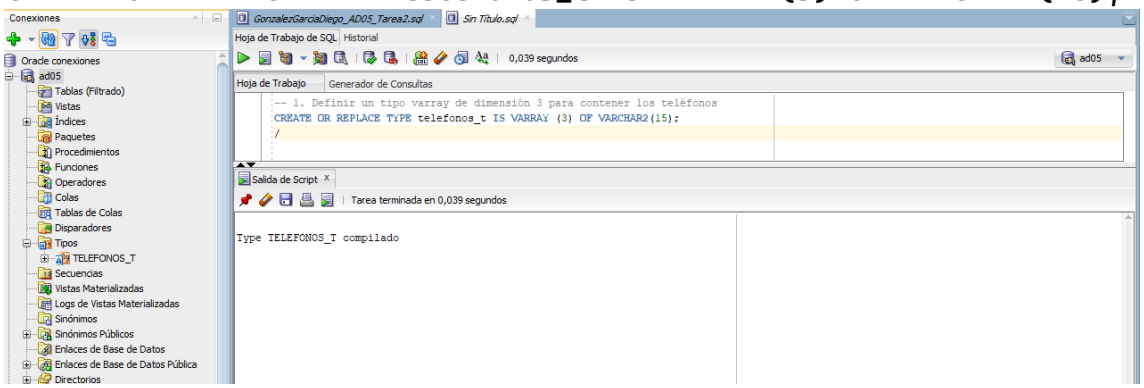
12 DE ENERO DE 2023

DIEGO GONZÁLEZ GARCÍA

Índice

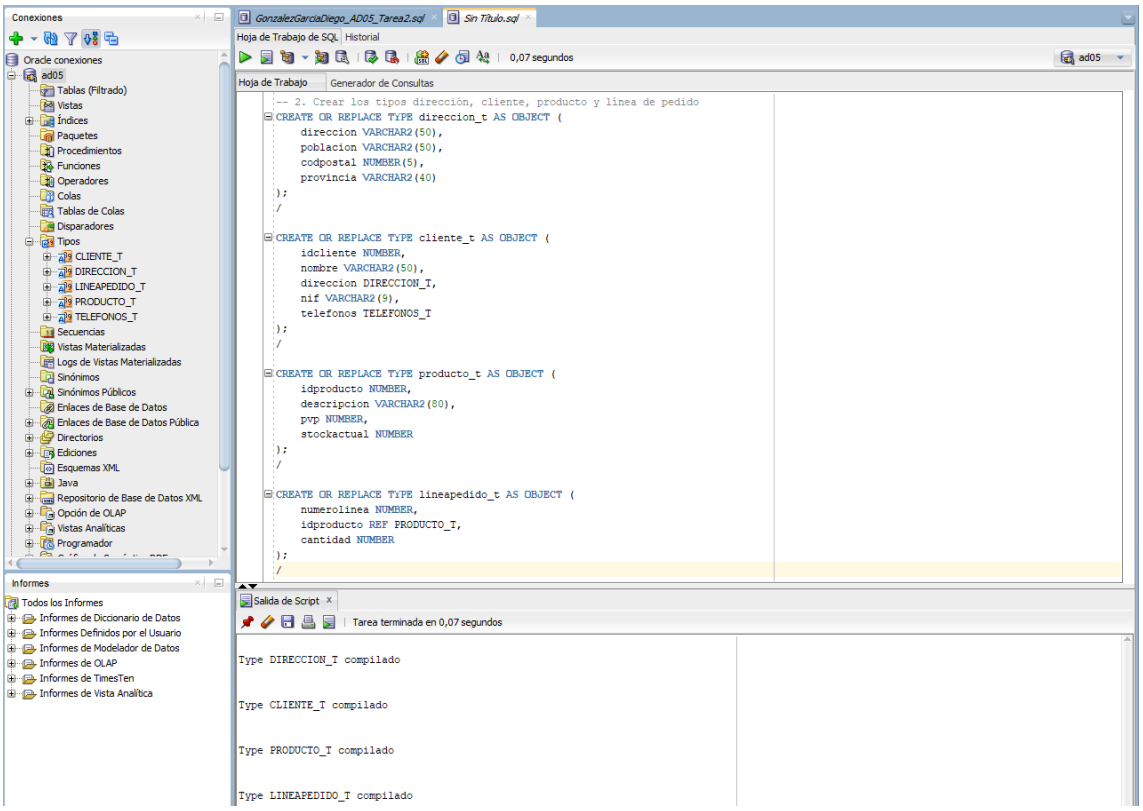
1. Definir un tipo varray de dimensión 3 para contener los teléfonos.

```
CREATE OR REPLACE TYPE telefonos_t IS VARRAY (3) OF VARCHAR2(15);
```



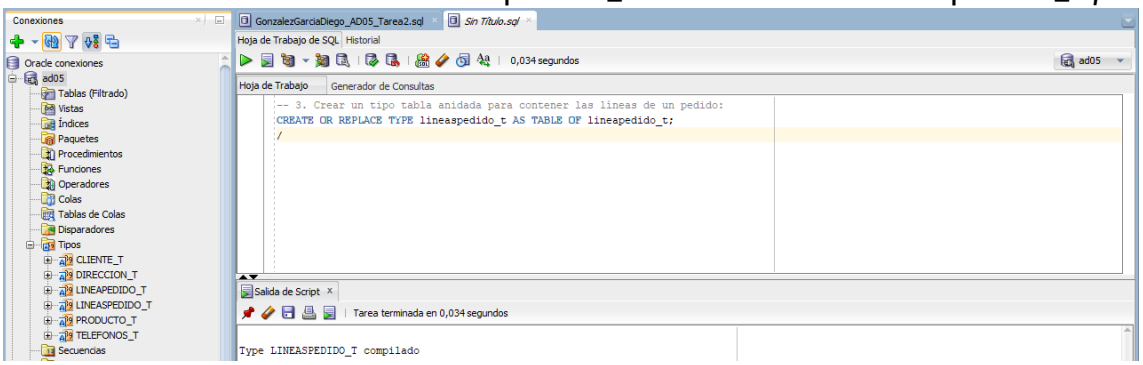
2. Crear los tipos dirección, cliente, producto y línea de pedido.

```
CREATE OR REPLACE TYPE direccion_t AS OBJECT (
    direccion VARCHAR2(50),
    poblacion VARCHAR2(50),
    codpostal NUMBER(5),
    provincia VARCHAR2(40)
);

CREATE OR REPLACE TYPE cliente_t AS OBJECT (
    idcliente NUMBER,
    nombre VARCHAR2(50),
    direccion DIRECCION_T,
    nif VARCHAR2(9),
    telefonos TELEFONOS_T
);

CREATE OR REPLACE TYPE producto_t AS OBJECT (
    idproducto NUMBER,
    descripcion VARCHAR2(80),
    pvp NUMBER,
    stockactual NUMBER
);

CREATE OR REPLACE TYPE lineapedido_t AS OBJECT (
    numerolinea NUMBER,
    idproducto REF PRODUCTO_T,
    cantidad NUMBER
);
```



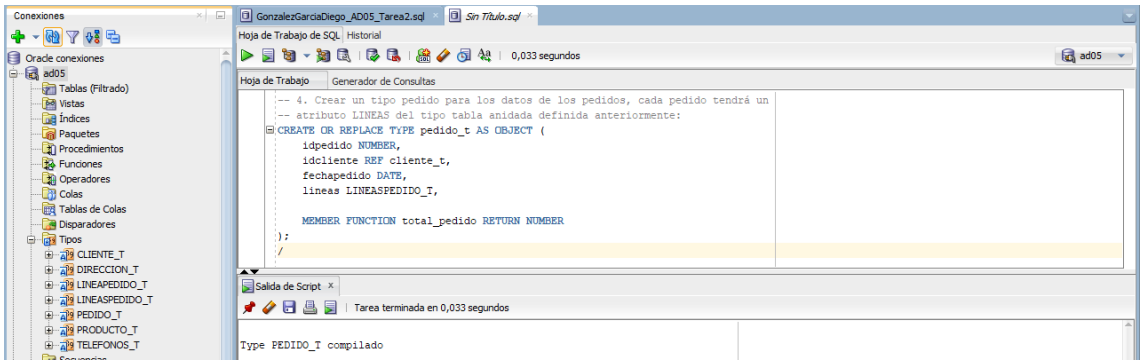
3. Crear un tipo tabla anidada para contener las líneas de un pedido.

```
CREATE OR REPLACE TYPE lineaspedido_t AS TABLE OF lineapedido_t;
```



4. Crear un tipo pedido para los datos de los pedidos, cada pedido tendrá un atributo LINEAS del tipo tabla anidada definida anteriormente.

```
CREATE OR REPLACE TYPE pedido_t AS OBJECT (
    idpedido NUMBER,
    idcliente REF cliente_t,
    fechapedido DATE,
    lineas LINEASPEDIDO_T
);
```



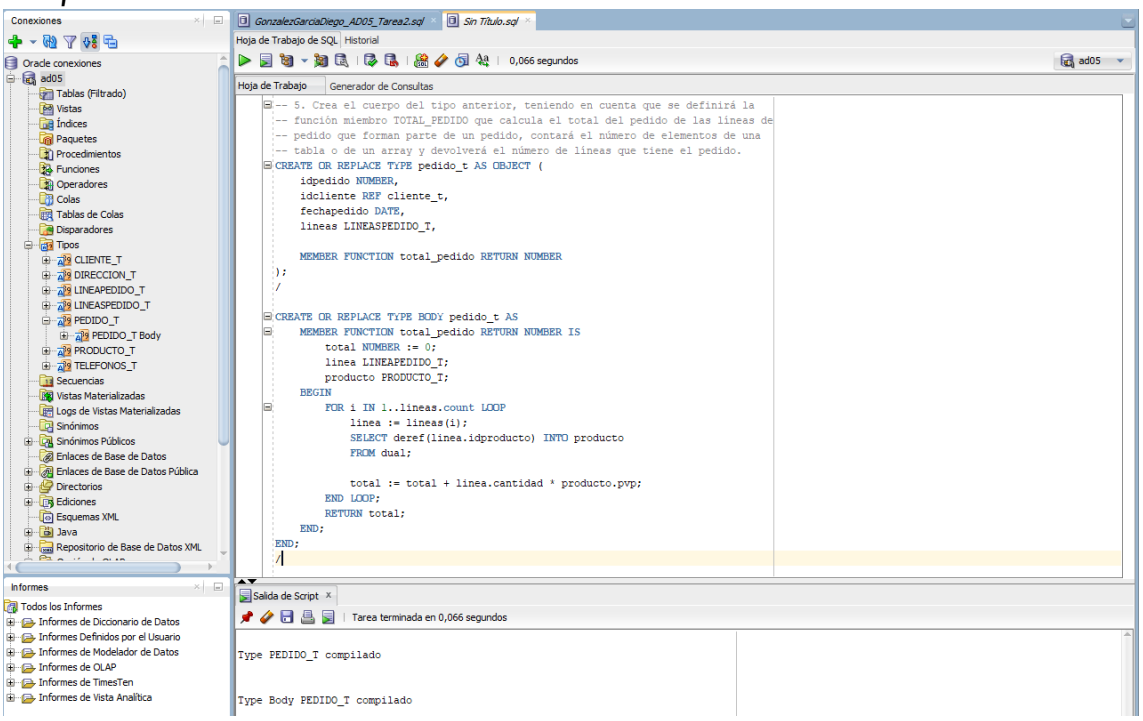
5. Crea el cuerpo del tipo anterior, teniendo en cuenta que se definirá la función miembro TOTAL_PEDIDO que calcula el total del pedido de las líneas de pedido que forman parte de un pedido, contará el número de elementos de una tabla o de un array y devolverá el número de líneas que tiene el pedido.

Para poder realizar este ejercicio, lo primero que debemos hacer es remplazar el tipo "pedido_t" que hemos definido previamente, para añadir la función miembro que se nos solicita. Una vez realizado este paso, podemos programar el cuerpo de "pedido_t".

El funcionamiento de la función es el siguiente: Se programa un bucle que va desde 1 hasta el total de la cuenta de las líneas que contiene el pedido y va calculando el subtotal de cada línea (cantidad de artículos por precio) y sumándoselo al total del pedido.

```
CREATE OR REPLACE TYPE pedido_t AS OBJECT (
    idpedido NUMBER,
    idcliente REF cliente_t,
    fechapedido DATE,
    lineas LINEASPEDIDO_T,

    MEMBER FUNCTION total_pedido RETURN NUMBER
);

CREATE OR REPLACE TYPE BODY pedido_t AS
    MEMBER FUNCTION total_pedido RETURN NUMBER IS
        total NUMBER := 0;
        linea LINEAPEDIDO_T;
        producto PRODUCTO_T;
    BEGIN
        FOR i IN 1..lineas.count LOOP
            linea := lineas(i);
            SELECT deref(linea.idproducto) INTO producto
            FROM dual;

            total := total + linea.cantidad * producto.pvp;
        END LOOP;
```

```
        RETURN total;
    END;
END;
```

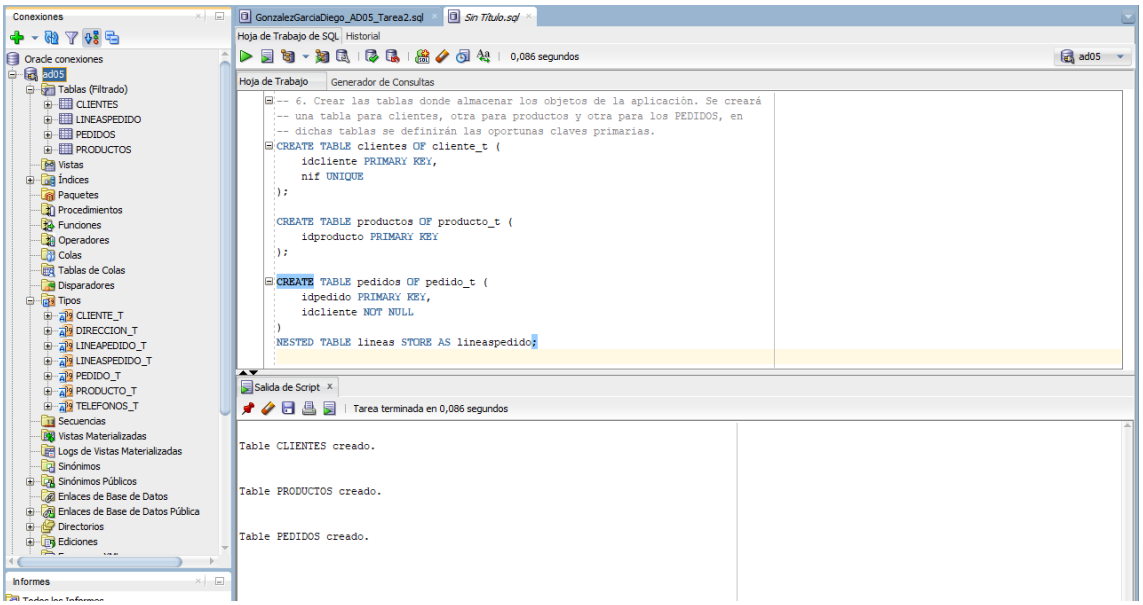


6. Crear las tablas donde almacenar los objetos de la aplicación. Se creará una tabla para clientes, otra para productos y otra para los PEDIDOS, en dichas tablas se definirán las oportunas claves primarias.

Además de las claves primarias, he definido algunas restricciones a algunos tipos de datos, ya que así constaban en el modelo de datos.

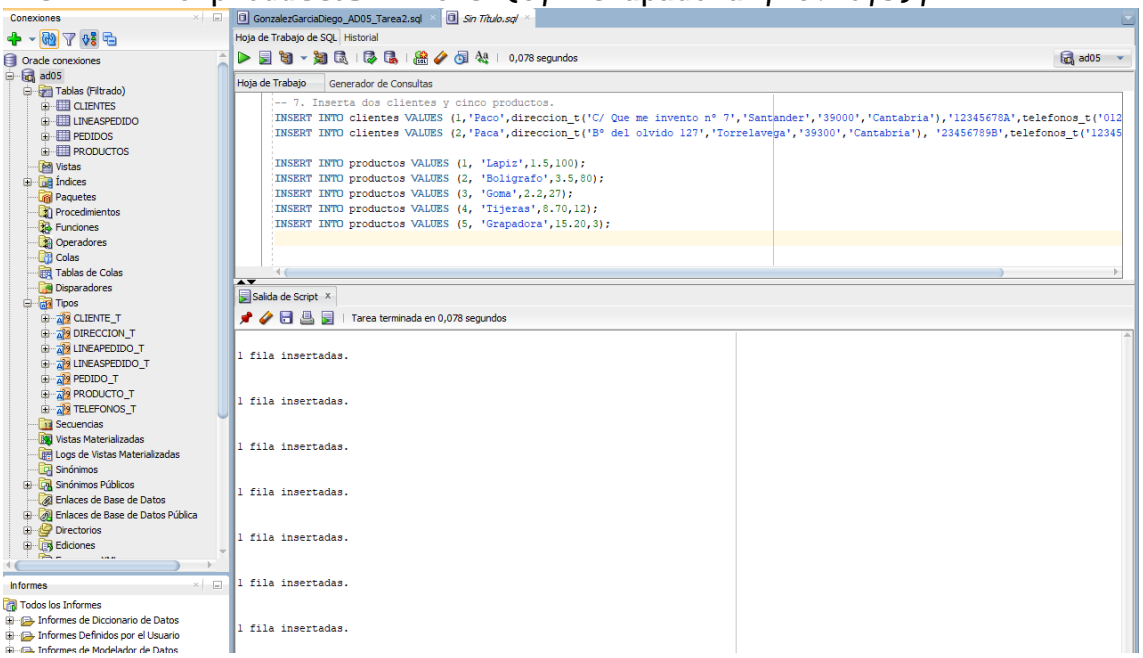
Para la tabla de pedidos también se incluye una tabla anidada para las líneas de pedido.

```
CREATE TABLE clientes OF cliente_t (
    idcliente PRIMARY KEY,
    nif UNIQUE
);

CREATE TABLE productos OF producto_t (
    idproducto PRIMARY KEY
);

CREATE TABLE pedidos OF pedido_t (
    idpedido PRIMARY KEY,
    idcliente NOT NULL
)
NESTED TABLE lineas STORE AS lineaspedido;
```



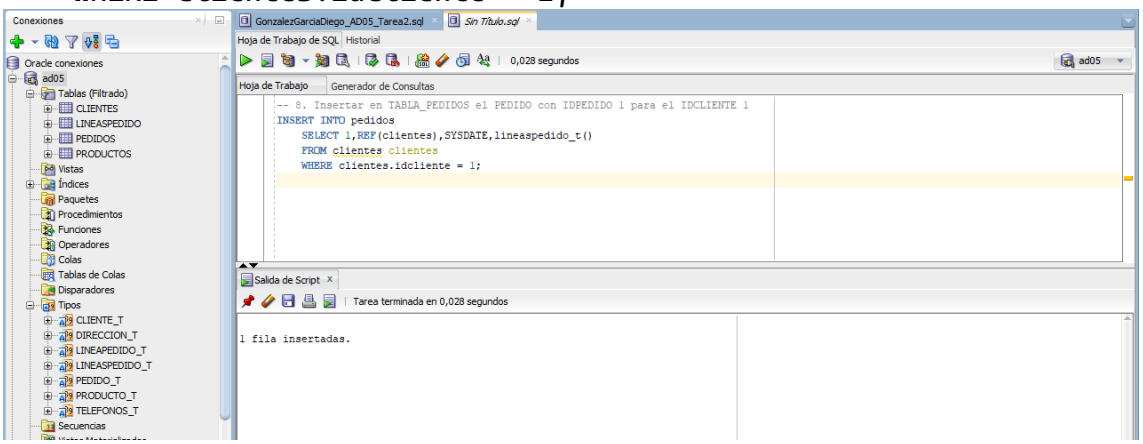
7. Inserta dos clientes y cinco productos.

```
INSERT INTO clientes VALUES (1,'Paco',direccion_t('C/ Que me invento nº 7','Santander','39000','Cantabria'),'12345678A',telefonos_t('012345678'));
INSERT INTO clientes VALUES (2,'Paca',direccion_t('Bº del olvido 127','Torrelavega','39300','Cantabria'), '23456789B',telefonos_t('123456789','987654321'));

INSERT INTO productos VALUES (1, 'Lapiz',1.5,100);
INSERT INTO productos VALUES (2, 'Boligrafo',3.5,80);
INSERT INTO productos VALUES (3, 'Goma',2.2,27);
INSERT INTO productos VALUES (4, 'Tijeras',8.70,12);
INSERT INTO productos VALUES (5, 'Grapadora',15.20,3);
```



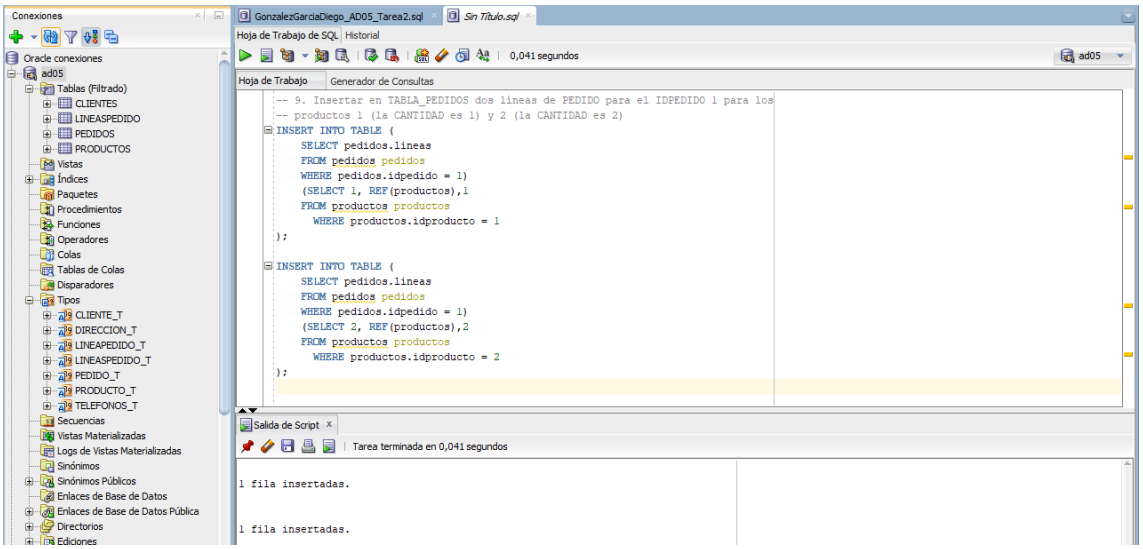
8. Insertar en TABLA_PEDIDOS el PEDIDO con IDPEDIDO 1 para el IDCLIENTE 1.

```
INSERT INTO pedidos
    SELECT 1,REF(clientes),SYSDATE,lineaspedido_t()
    FROM clientes clientes
    WHERE clientes.idcliente = 1;
```



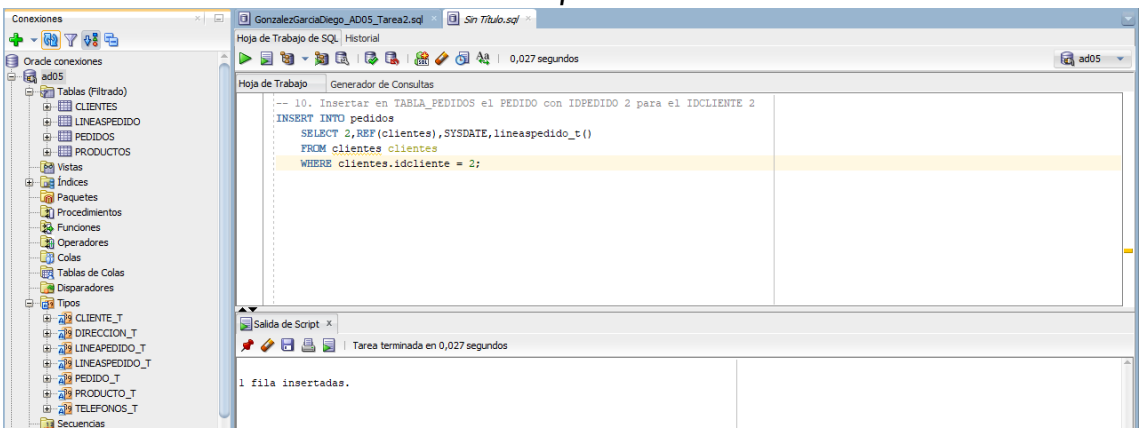
9. Insertar en TABLA_PEDIDOS dos líneas de PEDIDO para el IDPEDIDO 1 para los productos 1 (la CANTIDAD es 1) y 2 (la CANTIDAD es 2).

```
INSERT INTO TABLE (
    SELECT pedidos.lineas
    FROM pedidos pedidos
    WHERE pedidos.idpedido = 1)
    (SELECT 1, REF(productos),1
    FROM productos productos
      WHERE productos.idproducto = 1
);

INSERT INTO TABLE (
    SELECT pedidos.lineas
    FROM pedidos pedidos
    WHERE pedidos.idpedido = 1)
    (SELECT 2, REF(productos),2
    FROM productos productos
      WHERE productos.idproducto = 2
);
```



10. Insertar en TABLA_PEDIDOS el PEDIDO con IDPEDIDO 2 para el IDCLIENTE 2.

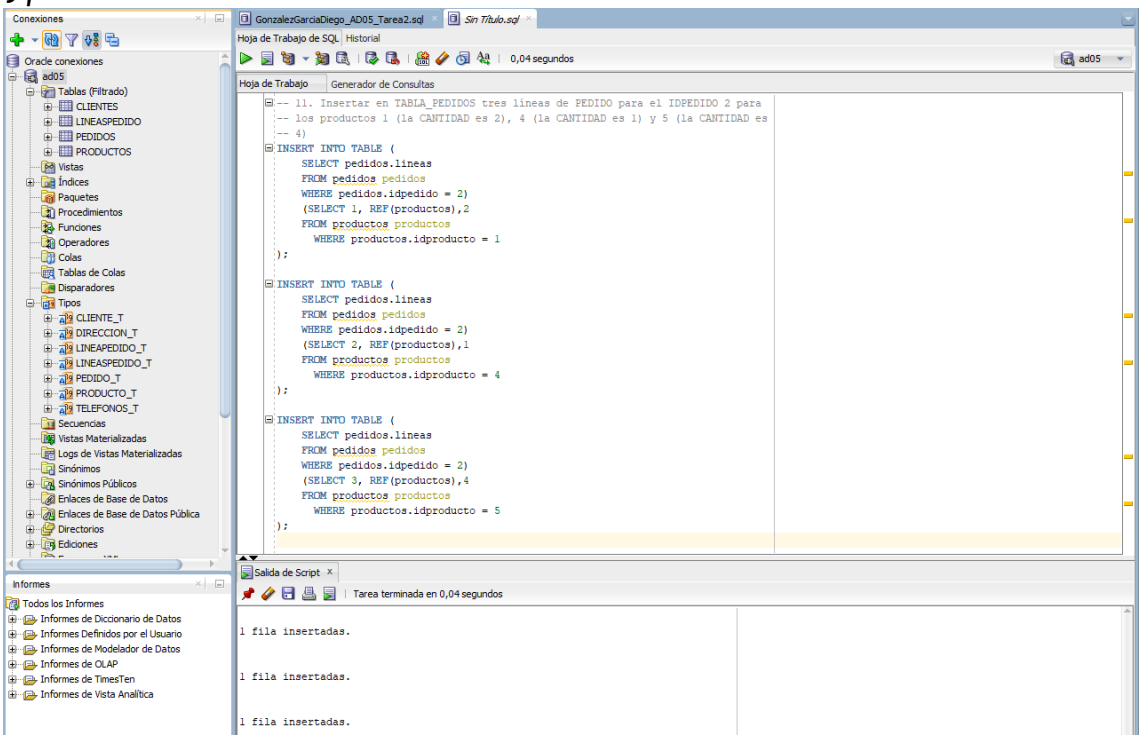
```
INSERT INTO pedidos
    SELECT 2,REF(clientes),SYSDATE,lineaspedido_t()
    FROM clientes clientes
    WHERE clientes.idcliente = 2;
```



11. Insertar en TABLA_PEDIDOS tres líneas de PEDIDO para el IDPEDIDO 2 para los productos 1 (la CANTIDAD es 2), 4 (la CANTIDAD es 1) y 5 (la CANTIDAD es 4).

```
INSERT INTO TABLE (
    SELECT pedidos.lineas
    FROM pedidos pedidos
    WHERE pedidos.idpedido = 2)
    (SELECT 1, REF(productos),2
    FROM productos productos
      WHERE productos.idproducto = 1
);
```

```
INSERT INTO TABLE (
    SELECT pedidos.lineas
    FROM pedidos pedidos
    WHERE pedidos.idpedido = 2)
    (SELECT 2, REF(productos),1
    FROM productos productos
      WHERE productos.idproducto = 4
);

INSERT INTO TABLE (
    SELECT pedidos.lineas
    FROM pedidos pedidos
    WHERE pedidos.idpedido = 2)
    (SELECT 3, REF(productos),4
    FROM productos productos
      WHERE productos.idproducto = 5
);
```



12. Realizar un procedimiento que recibiendo el identificador visualice los datos del PEDIDO.

Este ejercicio se comienza definiendo las variables que vamos a necesitar. Algunas de ellas las he definido a través de "%TYPE", ya que se refieren directamente a un campo de una tabla y de esta forma nos aseguramos de que su tipo de dato siempre será igual al contenido en la tabla.

Para las líneas del pedido se ha definido un cursor, con el que vamos sacando los datos de cada una, en base al "idpedido" que recibimos por parámetro.

Tras esto, mostramos los datos del pedido y a continuación vamos mostrando cada línea del pedido con un bucle. Finalizamos el muestreo de datos con el total del pedido que obtenemos a través de la función miembro que hemos definido en el ejercicio 5.

```
CREATE OR REPLACE PROCEDURE datospedido (idpedido_visualiza
pedidos.idpedido%TYPE) AS
    cliente CLIENTE_T;
    fecha pedidos.fechapedido%TYPE;
    precio productos.pvp%TYPE;
    subtotal NUMBER;
    total NUMBER;
    CURSOR curs IS
        SELECT numerolinea, DEREF(idproducto) producto, cantidad
        FROM THE (
            SELECT lineas
            FROM pedidos
            WHERE idpedido = idpedido_visualiza
            );

    BEGIN
        SELECT           DEREF(idcliente),           fechapedido,
pedidos.total_pedido()
        INTO cliente, fecha, total
        FROM pedidos pedidos
        WHERE idpedido = idpedido_visualiza;

        DBMS_OUTPUT.PUT_LINE('NUMERO    DE    PEDIDO:    '    ||
idpedido_visualiza);
        DBMS_OUTPUT.PUT_LINE('FECHA: ' || fecha);
        DBMS_OUTPUT.PUT_LINE('CLIENTE: ' || cliente.nombre);
        DBMS_OUTPUT.PUT_LINE('  NIF: ' || cliente.nif);
        DBMS_OUTPUT.PUT_LINE('           DIRECCION:     '     ||
cliente.direccion.direccion);
        DBMS_OUTPUT.PUT_LINE('                                  '  ||
cliente.direccion.codpostal || ', ' || cliente.direccion.poblacion
|| ', ' || cliente.direccion.provincia);

DBMS_OUTPUT.PUT_LINE('****************************************
****************');
        DBMS_OUTPUT.PUT_LINE('|  LINEA  |  DESCRIPCION  |  PRECIO
|  UN.  |  SUBTOTAL  |');

DBMS_OUTPUT.PUT_LINE('****************************************
****************');
        FOR i IN curs LOOP
            subtotal := i.cantidad * i.producto.pvp;
            DBMS_OUTPUT.PUT_LINE('|  ' || i.numerolinea || '_  |  '
|| i.producto.descripcion || '  |  ' || i.producto.pvp || ' €  |  '
|| i.cantidad || ' un.  |  ' || subtotal || ' €  |');
        END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('****************************************
****************');
        DBMS_OUTPUT.PUT_LINE('* TOTAL PEDIDO: ' || total || ' €');

DBMS_OUTPUT.PUT_LINE('****************************************
****************');
    END datospedido;
/

SET SERVEROUTPUT ON;
BEGIN
    datospedido(2);
END;
/
```