

AUTOR

Belén María Solís Valle

BASES DE DATOS DAW



Curso 2023/24

TAREA 06
Programación de Bases de Datos

Contenido

Caso práctico	3
¿Qué te pedimos que hagas?.....	3
APARTADO A:.....	4
A. Crear un procedimiento que reciba como parámetro el número de cesta y muestre por pantalla un listado con la información referente a la propia cesta (ID, nombre y apellidos del voluntario que la prepara, fecha de preparación, tipo de cesta y valor total e la misma). Cada uno de estos datos aparecerá en una línea diferente. A continuación, hay que mostrar el detalle de lo que contiene la cesta con el nombre del producto, la cantidad y la unidad de medida.....	4
• Para formatear la salida en cada una de las líneas se utiliza la función DBMS_OUTPUT.PUT_LINE que imprime en cada línea la información que se pasa como parámetros concatenando si es necesario varios valores o literales.....	4
• Para proporcionar espacios en blanco a la derecha o a la izquierda se puede utilizar las funciones RPAD y LPAD respectivamente. Puedes encontrar más información en los contenidos de la unidad 4 de este módulo donde se explican estas funciones en el apartdo 5.2. Funciones de cadena de caracteres para Oracle.....	4
APARTADO B:.....	7
B. Crear un procedimiento que reciba como parámetros el nombre y los apellidos de un socio y se actualice el tipo de ayuda que recibirá pudiendo ser un valor entre 1, 2 o 3 según el cálculo entre su renta y el número de familiares que tenga a su cargo.	7
El procedimiento tendrá que comparar el valor obtenido de la división entre la renta y el número de familiares que tiene a su cargo y realizar la siguiente actualización:	7
• Si el valor es inferior a 5000 entonces el tipo de ayuda que le pertenece a ese socio es 1.	7
• Si el valor está entre 5000 y 10000 entonces el tipo de ayuda que le pertenece a ese socio es 2.	7
• Si el valor es superior a 10000 entonces el tipo de ayuda que le pertenece a ese socio es 3.....	7
El procedimiento debe mostrar por pantalla el DNI del socio, su nombre completo, la renta, el número de familiares a su cargo, el valor calculado respecto a la renta y número de familiares y por último el tipo de ayuda que recibirá. Puedes guiarte con la siguiente salida por pantalla para que se parezca lo máximo posible:	7
• Para formatear la salida en cada una de las líneas se utiliza la función DBMS_OUTPUT.PUT_LINE que imprime en cada línea la información que se pasa como parámetros concatenando si es necesario varios valores o literales.....	7
APARTADO C:.....	11
C. Crear una función que reciba como parámetro el nombre de un producto y devuelva la cantidad total de ese producto que se ha repartido entre todas las cestas.	11
APARTADO D:.....	13
D. Crear un disparador (trigger) que se dispare cada vez que se prepare una nueva cesta para que se actualice los puntos acumulados del voluntario que la ha preparado con el valor de los puntos de dicha cesta.	13
• En este caso se deberá comprobar el funcionamiento del trigger y para ello deberás insertar algunos registros en la tabla CESTA (puedes dejar algunos valores a NULL, pero es necesario que indiques los puntos de la cesta y el DNI del voluntario que la prepara). Posteriormente debes consultar la tabla VOLUNTARIOS para comprobar que se ha actualizado el campo PuntosTotal.	13

Descripción de la tarea

Caso práctico

Siguiendo con los casos prácticos de las unidades anteriores, el equipo de Ana ya sabe realizar todo tipo de consultas y todo tipo de operaciones como la inserción, eliminación y la actualización de datos. Ahora de lo que se trata de es de aumentar las posibilidades que nos ofrecen las bases de datos aprendiendo a realizar pequeñas aplicaciones como procedimientos, funciones y triggers.

En esta unidad se ha explicado cómo hacer todo tipo de subprogramas tanto en MySQL como en Oracle. Hasta ahora en las unidades anteriores hemos trabajado con MySQL, pero es hora de aprender también a trabajar en otro SGBD como es Oracle. Por tanto, el equipo de Ana ha decidido adaptar el script de la BD de CESTA SOLIDARIA con el que han trabajado en las tareas 4 y 5 para que funcione en Oracle y realizar procedimientos, funciones y triggers con el lenguaje procedural PL-SQL.

¿Qué te pedimos que hagas?

Para automatizar ciertas operaciones en la BD y que puedan ser llamadas desde otras aplicaciones se desea programar algunas funcionalidades en la base de datos CESTA SOLIDARIA y tienes que implementar DOS procedimientos, UNA función y UN disparador.

Modelo Relacional

Socio (DNI, Nombre, apellidos, direccion, telefono, fecha_nacimiento, ingresos, numfam, ayuda, renta)

Voluntarios (DNI, Nombre, apellidos, direccion, telefono, fecha_nacimiento, puntostotal)

Cesta (Código, valor_total, tipo, fecharecoge, DNI_S puntos, fechaprepara, DNI_V)

Productos (referencia, nombre, valor, stock, medida)

Detalle_Cesta (Código_C, num_linea, cantidad, ref_p)

APARTADO A:

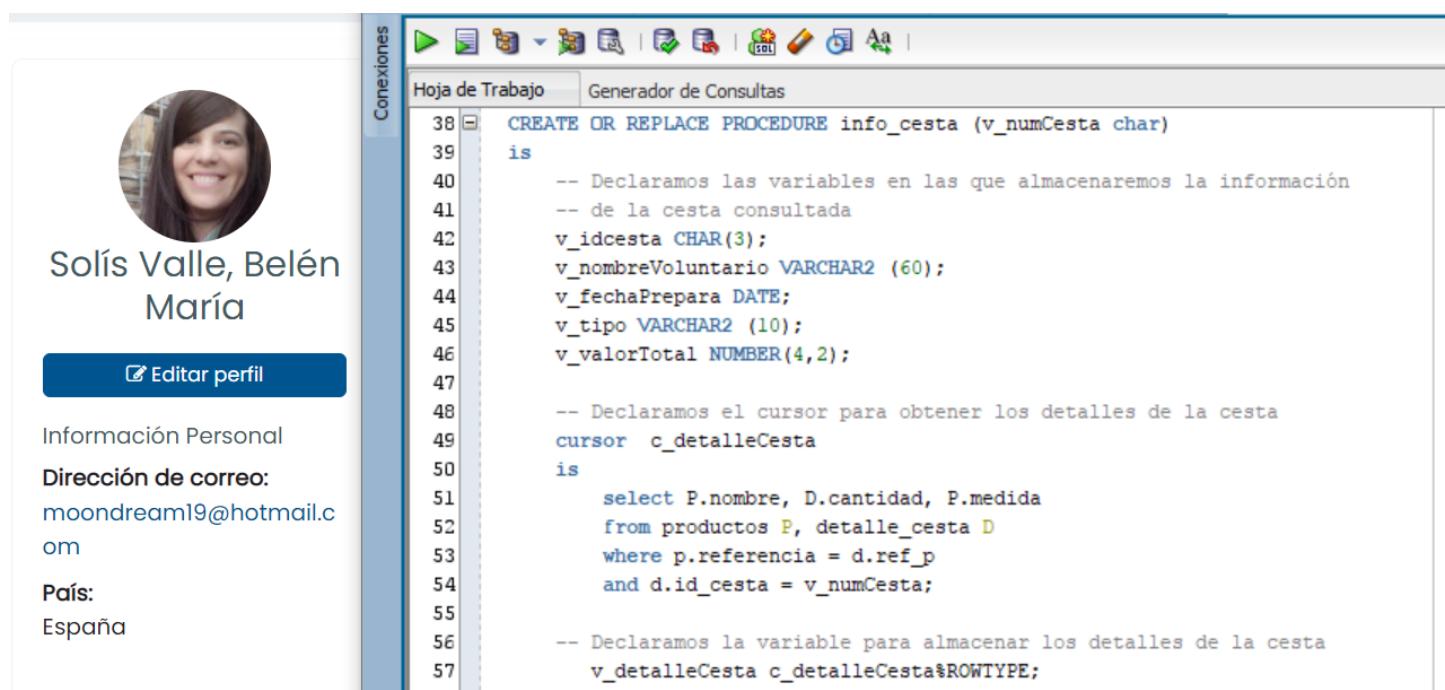
- A. Crear un procedimiento que reciba como parámetro el número de cesta y muestre por pantalla un listado con la información referente a la propia cesta (ID, nombre y apellidos del voluntario que la prepara, fecha de preparación, tipo de cesta y valor total de la misma). Cada uno de estos datos aparecerá en una línea diferente. A continuación, hay que mostrar el detalle de lo que contiene la cesta con el nombre del producto, la cantidad y la unidad de medida.

Puedes guiarte con la siguiente salida por pantalla para intentar reproducirla lo máximo posible:

```
Conectando a la base de datos tarea6_CESTA_2324.
ID Cesta : 002
Voluntario que prepara : Jose Luis Arce Navarro
Fecha de preparación : 01/01/23
Tipo de la Cesta : completa
Valor Total Cesta : 45,2
DETALLE DE LA CESTA
-----
PRODUCTO           CANTIDAD MEDIDA
-----
Pan                 3 Ud
Patatas             5 Kg
Azucar               4 Kg
Latas de conserva 10 Ud
Aceite Oliva        2 L
-----
El proceso ha terminado.
Desconectando de la base de datos tarea6_CESTA_2324.
```

- Para formatear la salida en cada una de las líneas se utiliza la función DBMS_OUTPUT.PUT_LINE que imprime en cada línea la información que se pasa como parámetros concatenando si es necesario varios valores o literales.
- Para proporcionar espacios en blanco a la derecha o a la izquierda se puede utilizar las funciones RPAD y LPAD respectivamente. Puedes encontrar más información en los contenidos de la unidad 4 de este módulo donde se explican estas funciones en el apartado 5.2. Funciones de cadena de caracteres para Oracle.

Declaramos las variables en las que almacenaremos la información de la cesta consultada, asimismo declaramos el cursor mediante el cual obtendremos los detalles de la cesta y su correspondiente variable. En este procedimiento, **info_cesta**, empleamos un cursor explícito para obtener los detalles de los productos incluidos en la cesta consultada.



The screenshot shows a user profile on the left with a photo of a woman, her name 'Solís Valle, Belén María', and a 'Editar perfil' button. On the right, the SQL Developer interface displays the PL/SQL code for the 'info_cesta' procedure. The code creates a procedure that declares variables for the basket ID, volunteer name, preparation date, basket type, and total value. It then declares a cursor 'c_detalleCesta' to select product details from the 'productos' and 'detalle_cesta' tables where the product reference matches the detail ID and the detail ID matches the basket ID. Finally, it declares a variable 'v_detalleCesta' of type 'c_detalleCesta%ROWTYPE'.

```

CREATE OR REPLACE PROCEDURE info_cesta (v_numCesta char)
is
    -- Declaramos las variables en las que almacenaremos la información
    -- de la cesta consultada
    v_idcesta CHAR(3);
    v_nombreVoluntario VARCHAR2 (60);
    v_fechaPrepara DATE;
    v_tipo VARCHAR2 (10);
    v_valorTotal NUMBER(4,2);

    -- Declaramos el cursor para obtener los detalles de la cesta
    cursor c_detalleCesta
    is
        select P.nombre, D.cantidad, P.medida
        from productos P, detalle_cesta D
        where p.referencia = d.ref_p
        and d.id_cesta = v_numCesta;

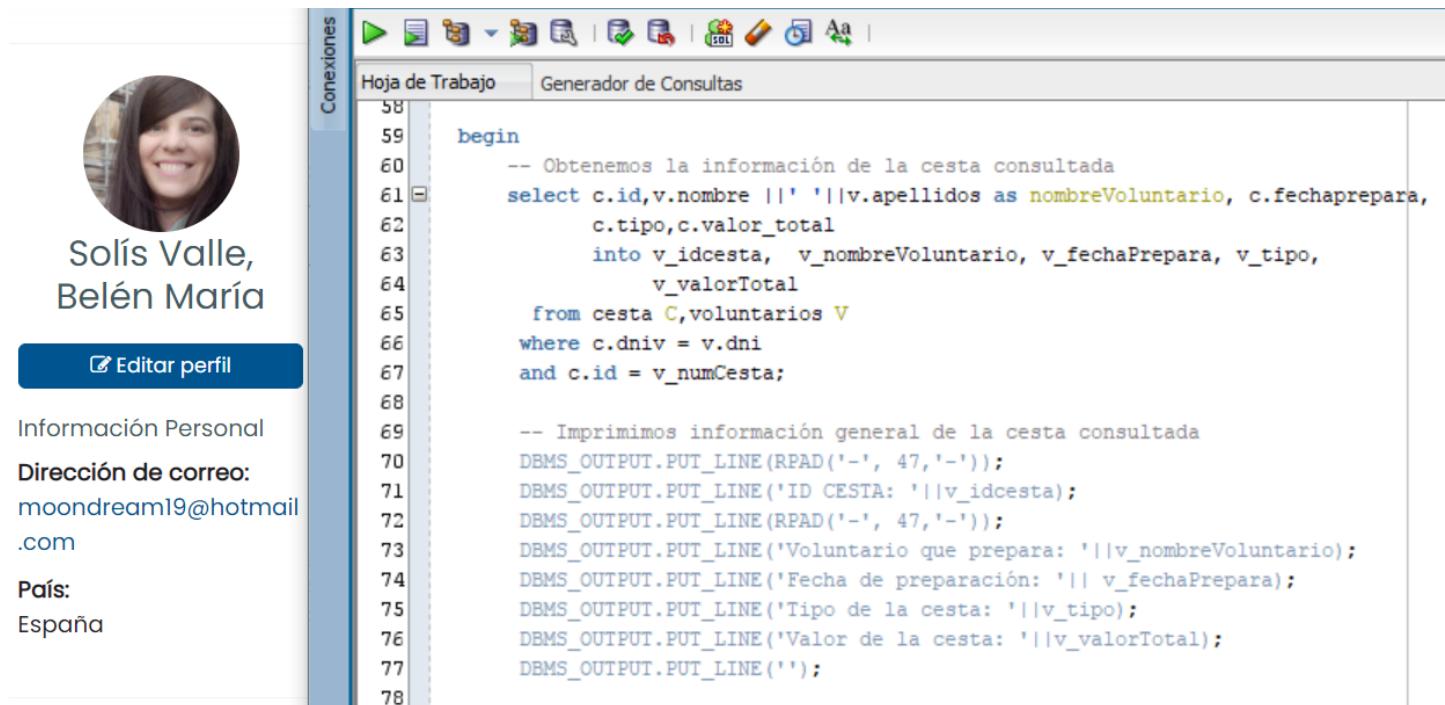
    -- Declaramos la variable para almacenar los detalles de la cesta
    v_detalleCesta c_detalleCesta%ROWTYPE;

```

Obtenemos la información de la cesta consultada mediante un cursor implícito por el cual obtenemos los valores de la cesta consultada: id de la cesta, nombre completo del voluntario, fecha de preparación de la cesta, tipo de cesta y valor total de la cesta.

Cuando una consulta SQL devuelve una sola fila, no es necesario declarar un cursor explícito como en este caso. Podemos asignar los valores directamente a las variables. La información de la cesta se recupera utilizando una consulta SQL y se almacena en las variables v_idcesta, v_nombreVoluntario, v_fechaPrepara, v_tipo y v_valorTotal.

Tras lo cual imprimimos la información general de la cesta consultada.



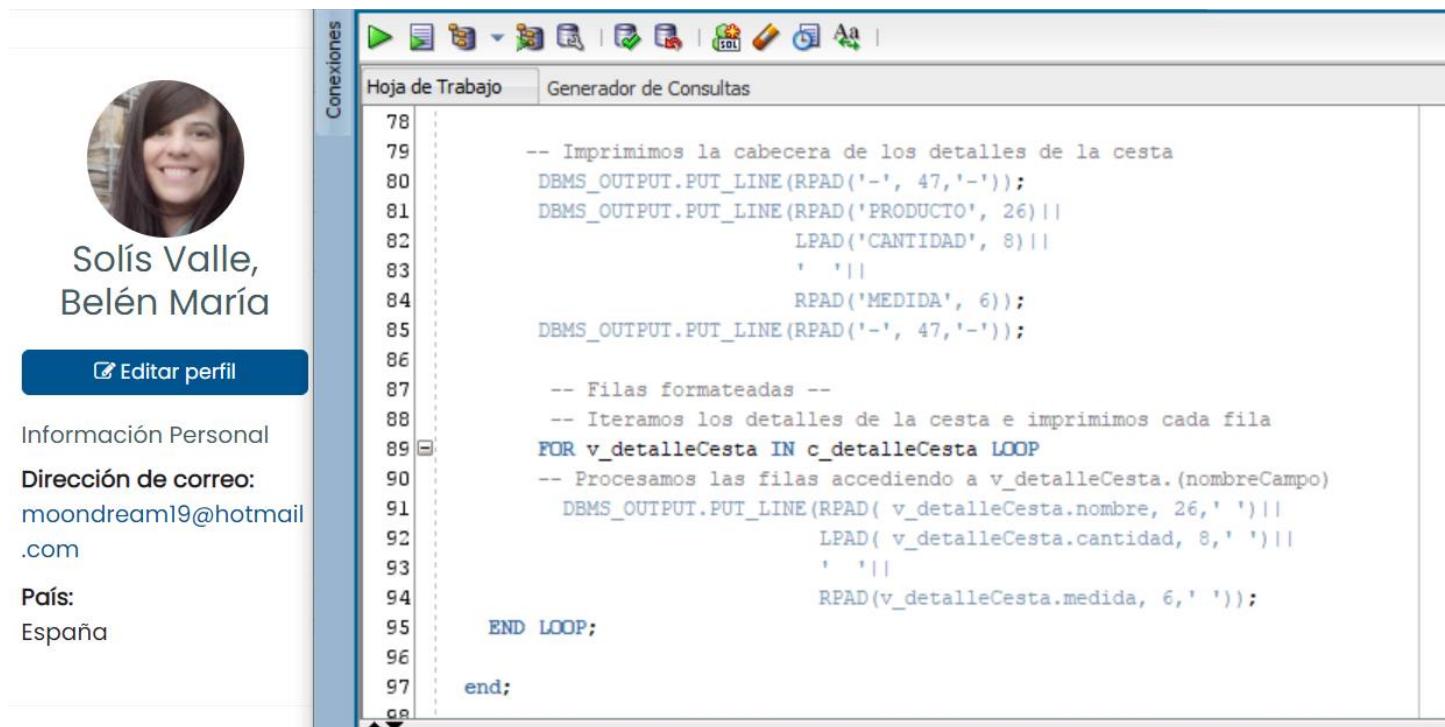
```

58 begin
59     -- Obtenemos la información de la cesta consultada
60     select c.id,v.nombre ||' '||v.apellidos as nombreVoluntario, c.fechaPrepara,
61         c.tipo,c.valor_total
62         into v_idcesta, v_nombreVoluntario, v_fechaPrepara, v_tipo,
63             v_valorTotal
64         from cesta C,voluntarios V
65         where c.dniv = v.dni
66         and c.id = v_numCesta;
67
68         -- Imprimimos información general de la cesta consultada
69         DBMS_OUTPUT.PUT_LINE(RPAD('-', 47, '-'));
70         DBMS_OUTPUT.PUT_LINE('ID CESTA: '||v_idcesta);
71         DBMS_OUTPUT.PUT_LINE(RPAD('-', 47, '-'));
72         DBMS_OUTPUT.PUT_LINE('Voluntario que prepara: '||v_nombreVoluntario);
73         DBMS_OUTPUT.PUT_LINE('Fecha de preparación: '|| v_fechaPrepara);
74         DBMS_OUTPUT.PUT_LINE('Tipo de la cesta: '||v_tipo);
75         DBMS_OUTPUT.PUT_LINE('Valor de la cesta: '||v_valorTotal);
76         DBMS_OUTPUT.PUT_LINE('');
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

```

Imprimimos la cabecera de los detalles de la cesta. El cursor explícito que habíamos declarado selecciona los nombres de los productos, las cantidades y las medidas de la tabla productos y detalle_cesta. **Estos detalles se pueden iterar en un bucle FOR para recorrer cada fila del cursor y procesar la información.**

Este bucle FOR itera sobre cada fila devuelta por el cursor c_detalleCesta y procesa la información de cada producto incluido en la cesta, imprimiendo el nombre del producto, la cantidad y la medida en la salida.

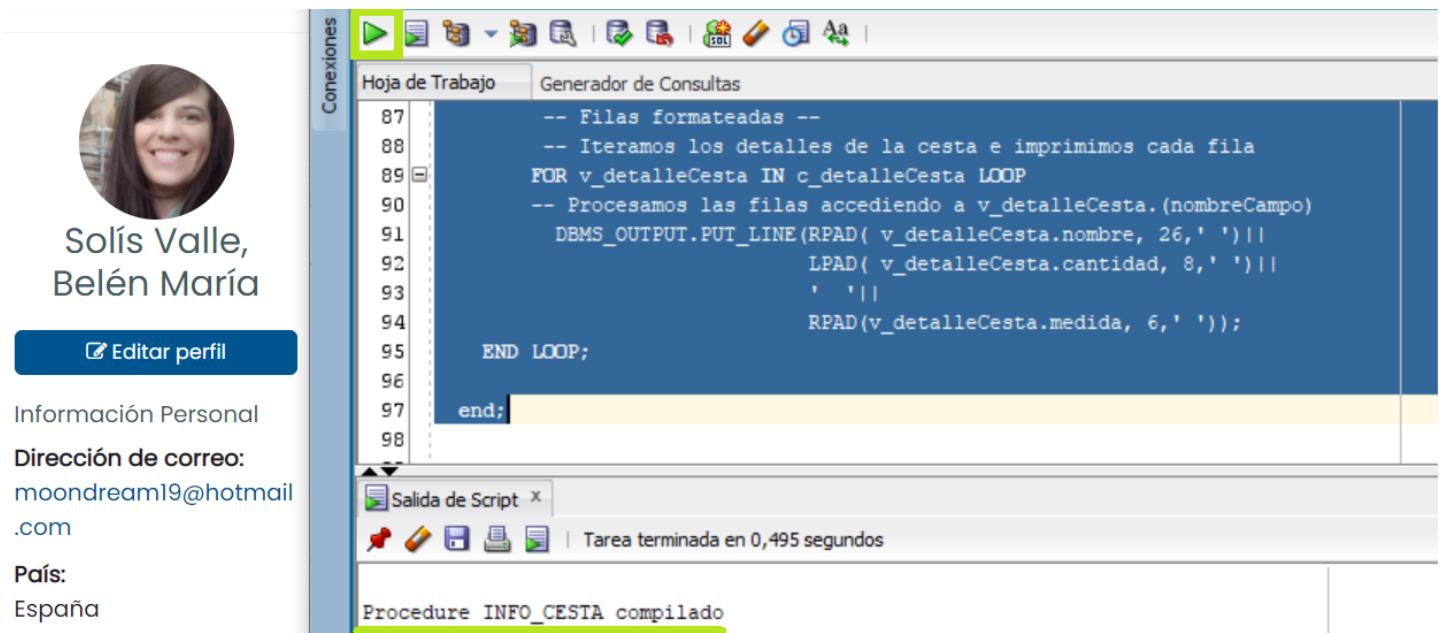


```

78
79         -- Imprimimos la cabecera de los detalles de la cesta
80         DBMS_OUTPUT.PUT_LINE(RPAD('-', 47, '-'));
81         DBMS_OUTPUT.PUT_LINE(RPAD('PRODUCTO', 26) ||
82                             LPAD('CANTIDAD', 8) ||
83                             ' ' ||
84                             RPAD('MEDIDA', 6));
85         DBMS_OUTPUT.PUT_LINE(RPAD('-', 47, '-'));
86
87         -- Filas formateadas --
88         -- Iteramos los detalles de la cesta e imprimimos cada fila
89         FOR v_detalleCesta IN c_detalleCesta LOOP
90             -- Procesamos las filas accediendo a v_detalleCesta.(nombreCampo)
91             DBMS_OUTPUT.PUT_LINE(RPAD( v_detalleCesta.nombre, 26, ' ') ||
92                                 LPAD( v_detalleCesta.cantidad, 8, ' ') ||
93                                 ' ' ||
94                                 RPAD(v_detalleCesta.medida, 6, ' '));
95         END LOOP;
96
97
98

```

Compilamos el procedimiento



Solís Valle, Belén María

Editar perfil

Información Personal

Dirección de correo:
moondream19@hotmail.com

País:
España

Connexiones

Hoja de Trabajo Generador de Consultas

```

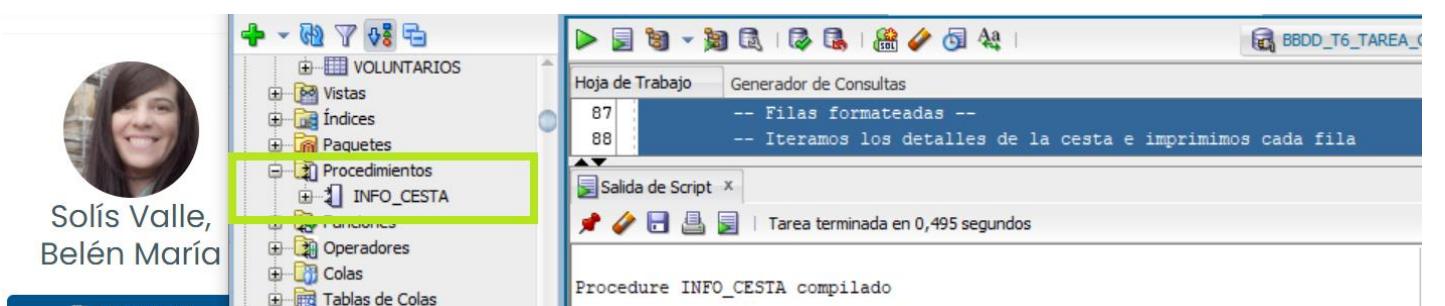
87      -- Filas formateadas --
88      -- Iteramos los detalles de la cesta e imprimimos cada fila
89      FOR v_detalleCesta IN c_detalleCesta LOOP
90          -- Procesamos las filas accediendo a v_detalleCesta.(nombreCampo)
91          DBMS_OUTPUT.PUT_LINE(RPAD( v_detalleCesta.nombre, 26, ' ') ||
92                                LPAD( v_detalleCesta.cantidad, 8, ' ') ||
93                                ' ' ||
94                                RPAD(v_detalleCesta.medida, 6, ' '));
95      END LOOP;
96
97  end;
98

```

Salida de Script x | Tarea terminada en 0,495 segundos

Procedure INFO_CESTA compilado

Tras la compilación nos aparece nuestro nuevo procedimiento '**INFO_CESTA**'.



Solís Valle, Belén María

Editar perfil

VOLUNTARIOS

- Vistas
- Índices
- Paquetes
- Procedimientos
 - INFO_CESTA**
- Funciones
- Operadores
- Colas
- Tablas de Colas

Connexiones

Hoja de Trabajo Generador de Consultas

```

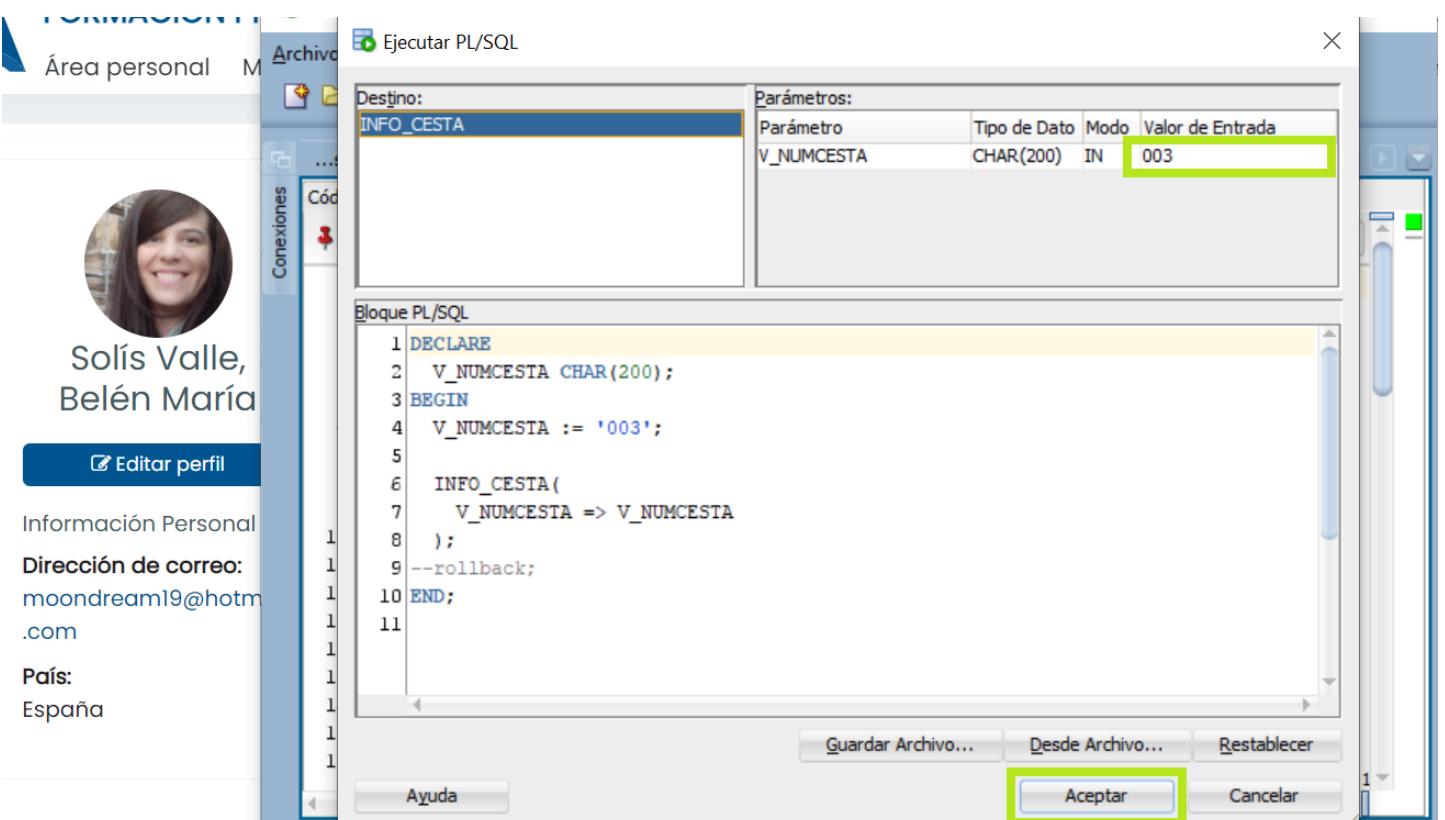
87      -- Filas formateadas --
88      -- Iteramos los detalles de la cesta e imprimimos cada fila

```

Salida de Script x | Tarea terminada en 0,495 segundos

Procedure INFO_CESTA compilado

Ejecutamos el procedimiento introduciendo el valor de entrada '**003**'



Solís Valle, Belén María

Editar perfil

Información Personal

Dirección de correo:
moondream19@hotmail.com

País:
España

Área personal

Ejecutar PL/SQL

Destino: INFO_CESTA

Parámetro	Tipo de Dato	Modo	Valor de Entrada
V_NUMCESTA	CHAR(200)	IN	003

Bloque PL/SQL

```

1 DECLARE
2     V_NUMCESTA CHAR(200);
3 BEGIN
4     V_NUMCESTA := '003';
5
6     INFO_CESTA(
7         V_NUMCESTA => V_NUMCESTA
8     );
9     --rollback;
10    END;
11

```

Guardar Archivo... Desde Archivo... Restablecer

Aceptar Cancelar

Tras la ejecución del procedimiento obtenemos la información de la cesta y su correspondiente detalle

```

Ejecutando: IdeConnections%23BBDD_T6_TAREA_CESTA_SOLIDARIA.jpr - Log

Conectando a la base de datos BBDD_T6_TAREA_CESTA_SOLIDARIA.

ID CESTA: 003

Voluntario que prepara: Patricia Fernandez Ruiz
Fecha de preparación: 15/02/23
Tipo de la cesta: estandar
Valor de la cesta: 36

PRODUCTO          CANTIDAD MEDIDA
-----
Aceite Girasol      4 L
Leche              5 L
Latas de conserva   2 Ud
Azucar              9 Kg
Patatas            3 Kg

El proceso ha terminado.
Desconectando de la base de datos BBDD_T6_TAREA_CESTA_SOLIDARIA.

Ejecutando: IdeConnections%23BD_... x Ejecutando: IdeConnections%23BBDD_T6_TAREA_CESTA_SOLIDARIA.jpr - Log
Ejecutando: IdeConnections%23BBDD_T6_TAREA_CESTA_SOLIDARIA.jpr - Log
Se ha guardado: D:\Bases de datos 2023-2024\BBDD_UID6\BBDD_T6_TAREA_PROCEDIMIENTO_A.sql

```

APARTADO B:

- B. Crear un procedimiento que reciba como parámetros el nombre y los apellidos de un socio y se actualice el tipo de ayuda que recibirá pudiendo ser un valor entre 1, 2 o 3 según el cálculo entre su renta y el número de familiares que tenga a su cargo.

El procedimiento tendrá que comparar el valor obtenido de la división entre la renta y el número de familiares que tiene a su cargo y realizar la siguiente actualización:

- Si el valor es inferior a 5000 entonces el tipo de ayuda que le pertenece a ese socio es 1.
- Si el valor está entre 5000 y 10000 entonces el tipo de ayuda que le pertenece a ese socio es 2.
- Si el valor es superior a 10000 entonces el tipo de ayuda que le pertenece a ese socio es 3.

El procedimiento debe mostrar por pantalla el DNI del socio, su nombre completo, la renta, el número de familiares a su cargo, el valor calculado respecto a la renta y número de familiares y por último el tipo de ayuda que recibirá. Puedes guiarte con la siguiente salida por pantalla para que se parezca lo máximo posible:

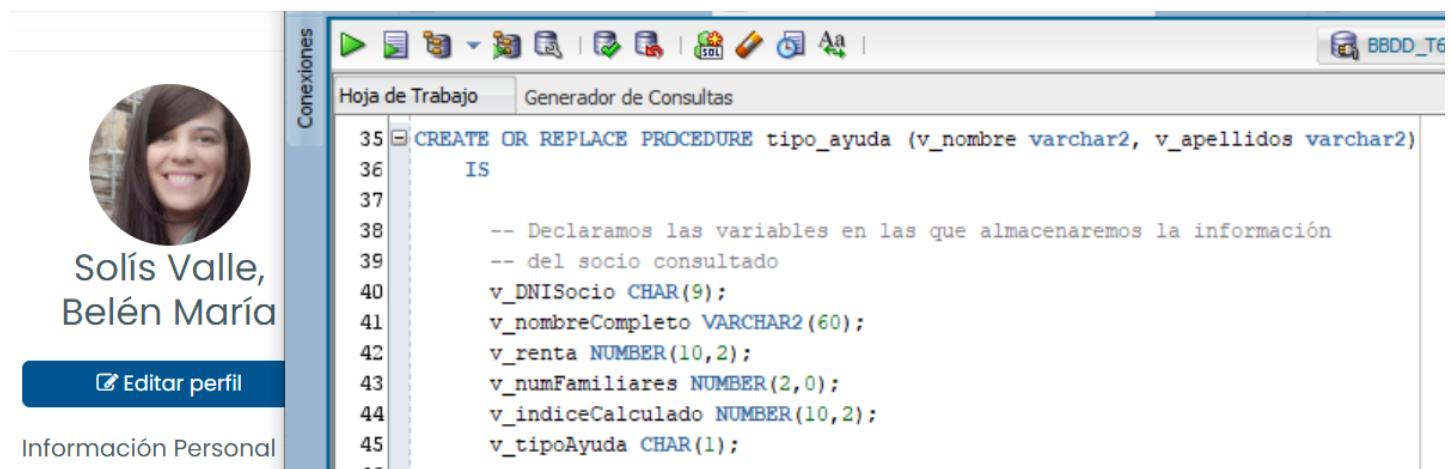
```

Conectando a la base de datos tarea6_CESTA_2324.
El socio con DNI 98452175H se llama Eduardo Lopez Lopez
Tiene una renta de 25000 y tiene 3 familiares a su cargo.
Según el índice calculado es 8333,33 y se actualiza el tipo de ayuda a 2
El proceso ha terminado.
Desconectando de la base de datos tarea6_CESTA_2324.

```

- Para formatear la salida en cada una de las líneas se utiliza la función DBMS_OUTPUT.PUT_LINE que imprime en cada línea la información que se pasa como parámetros concatenando si es necesario varios valores o literales.

Declaramos las variables en las que almacenaremos la información del socio consultado: **v_DNISocio**, **v_nombreCompleto**, **v_renta**, **v_numFamiliares**, **v_indiceCalculado** y **v_tipoAyuda**.



Solís Valle,
Belén María

Editar perfil

Información Personal

```

35 CREATE OR REPLACE PROCEDURE tipo_ayuda (v_nombre varchar2, v_apellidos varchar2)
36   IS
37
38   -- Declaramos las variables en las que almacenaremos la información
39   -- del socio consultado
40   v_DNISocio CHAR(9);
41   v_nombreCompleto VARCHAR2(60);
42   v_renta NUMBER(10,2);
43   v_numFamiliares NUMBER(2,0);
44   v_indiceCalculado NUMBER(10,2);
45   v_tipoAyuda CHAR(1);

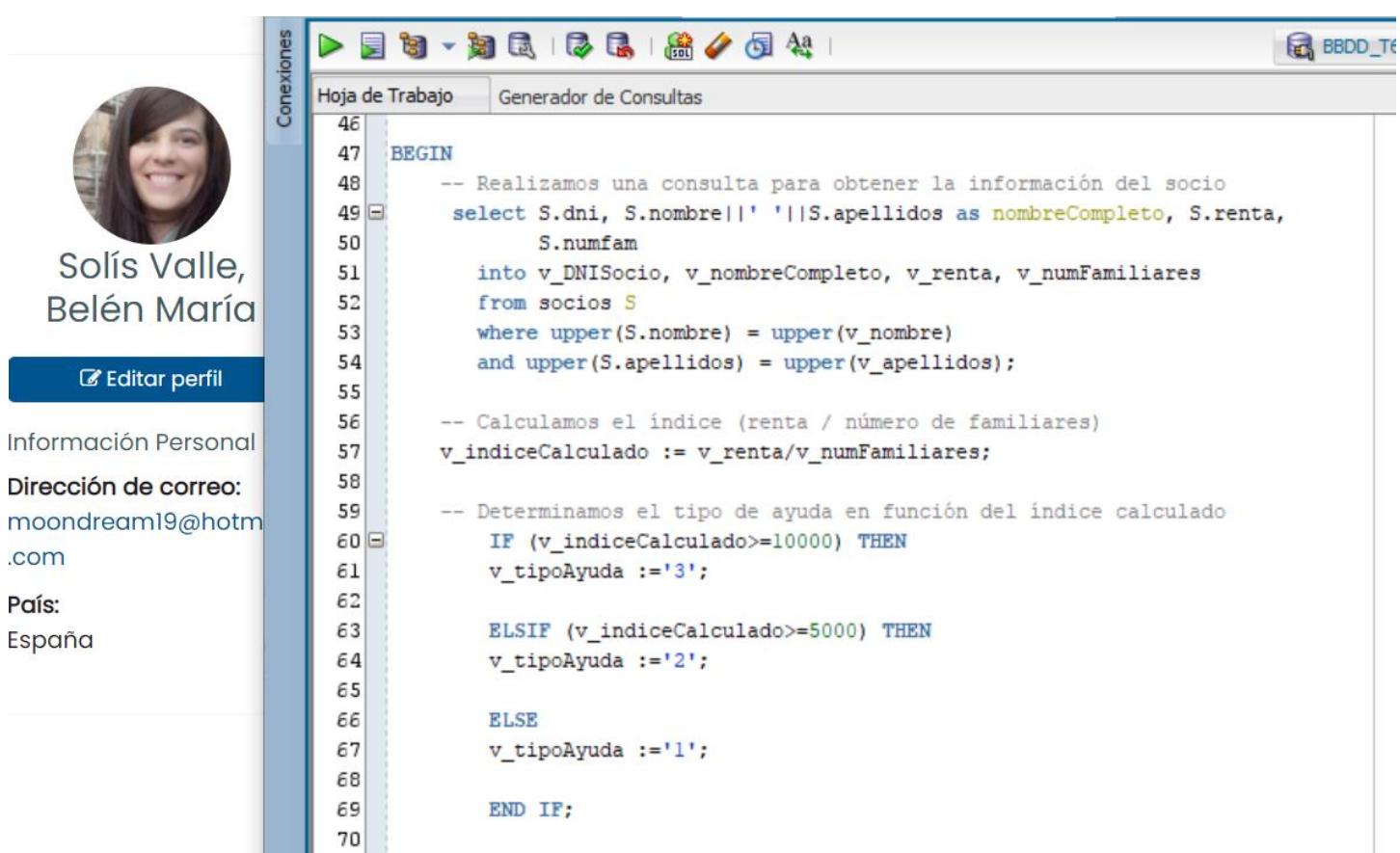
```

Realizamos una consulta para obtener la información del socio. Esta consulta busca un socio en la tabla socios cuyo nombre y apellidos coincidan con los proporcionados como parámetros (**v_nombre** y **v_apellidos**). Tras lo cual almacenamos el resultado en las variables **v_DNISocio**, **v_nombreCompleto**, **v_renta** y **v_numFamiliares**.

Tras lo cual **calculamos el índice de renta** para ello dividimos la renta del socio entre el número de familiares y almacenamos el resultado en la variable **v_indiceCalculado**.

A continuación, **determinamos el tipo de ayuda** en función del índice que hemos calculado:

- Si el valor es inferior a 5000 entonces el tipo de ayuda que le pertenece a ese socio es 1.
- Si el valor está entre 5000 y 10000 entonces el tipo de ayuda que le pertenece a ese socio es 2.
- Si el valor es superior a 10000 entonces el tipo de ayuda que le pertenece a ese socio es 3.



Solís Valle,
Belén María

Editar perfil

Información Personal

Dirección de correo:
moondream19@hotmail.com

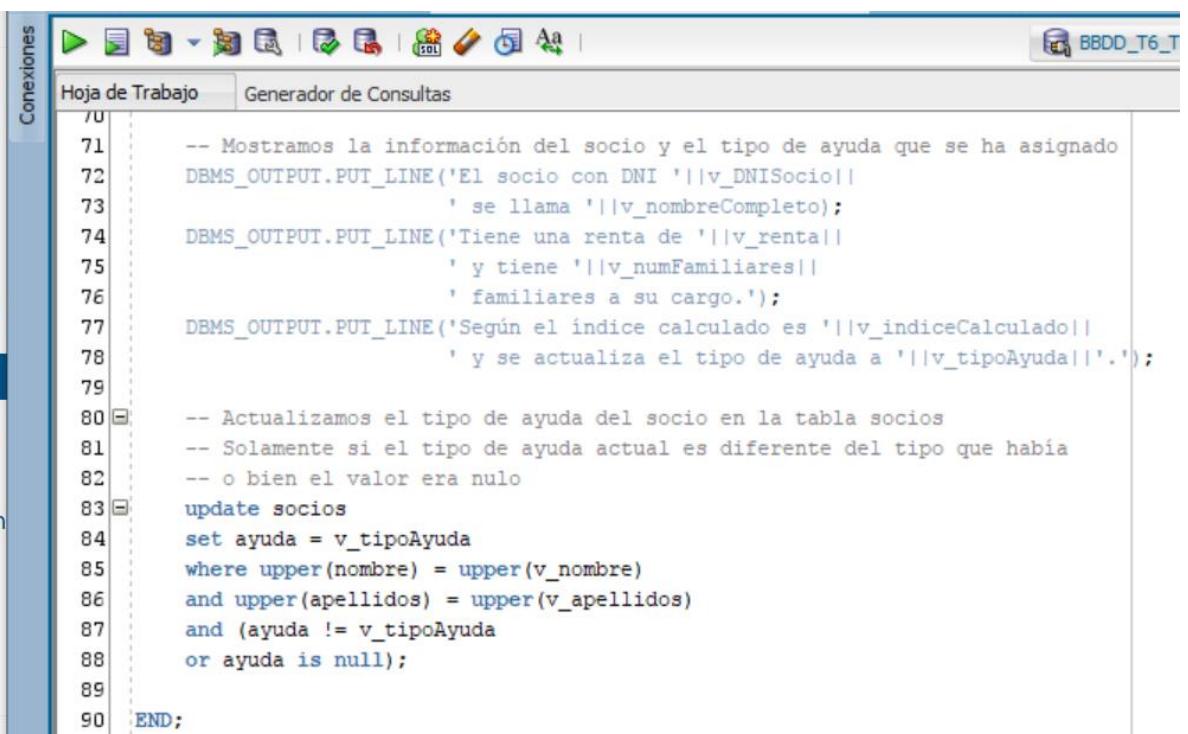
País:
España

```

46 BEGIN
47
48   -- Realizamos una consulta para obtener la información del socio
49   select S.dni, S.nombre||' '||S.apellidos as nombreCompleto, S.renta,
50         S.numfam
51   into v_DNISocio, v_nombreCompleto, v_renta, v_numFamiliares
52   from socios S
53   where upper(S.nombre) = upper(v_nombre)
54     and upper(S.apellidos) = upper(v_apellidos);
55
56   -- Calculamos el índice (renta / número de familiares)
57   v_indiceCalculado := v_renta/v_numFamiliares;
58
59   -- Determinamos el tipo de ayuda en función del índice calculado
60   IF (v_indiceCalculado>=10000) THEN
61     v_tipoAyuda :='3';
62
63   ELSIF (v_indiceCalculado>=5000) THEN
64     v_tipoAyuda :='2';
65
66   ELSE
67     v_tipoAyuda :='1';
68
69   END IF;
70

```

Mostramos la información del socio y el tipo de ayuda que se le ha asignado en función al índice de su renta. Tras lo cual actualizamos la tabla socios y establecemos el tipo de ayuda a percibir. Esta actualización solamente se realizará si el tipo de ayuda actual es diferente al tipo de ayuda asignado o bien el valor del tipo de ayuda era nulo.



```

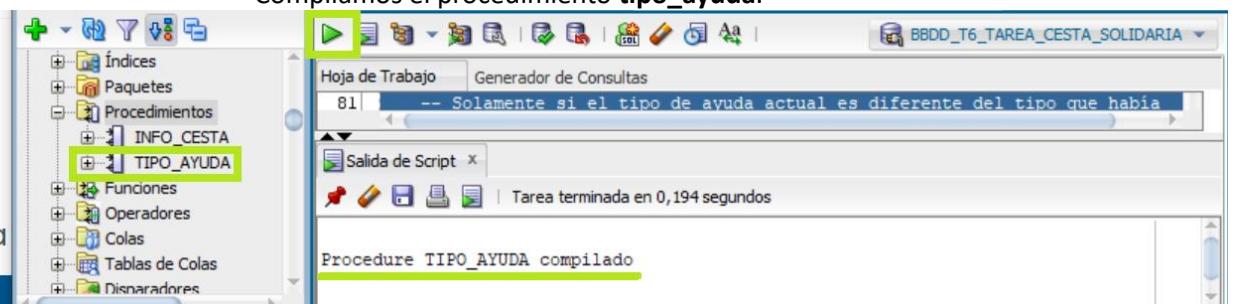
-- Mostramos la información del socio y el tipo de ayuda que se ha asignado
DBMS_OUTPUT.PUT_LINE('El socio con DNI'||v_DNISocio||
                      ' se llama'||v_nombreCompleto);
DBMS_OUTPUT.PUT_LINE('Tiene una renta de'||v_renta||
                      ' y tiene'||v_numFamiliares||
                      ' familiares a su cargo.');
DBMS_OUTPUT.PUT_LINE('Según el índice calculado es'||v_indiceCalculado||
                      ' y se actualiza el tipo de ayuda a'||v_tipoAyuda||');

-- Actualizamos el tipo de ayuda del socio en la tabla socios
-- Solamente si el tipo de ayuda actual es diferente del tipo que había
-- o bien el valor era nulo
update socios
set ayuda = v_tipoAyuda
where upper(nombre) = upper(v_nombre)
and upper(apellidos) = upper(v_apellidos)
and (ayuda != v_tipoAyuda
or ayuda is null);

END;

```

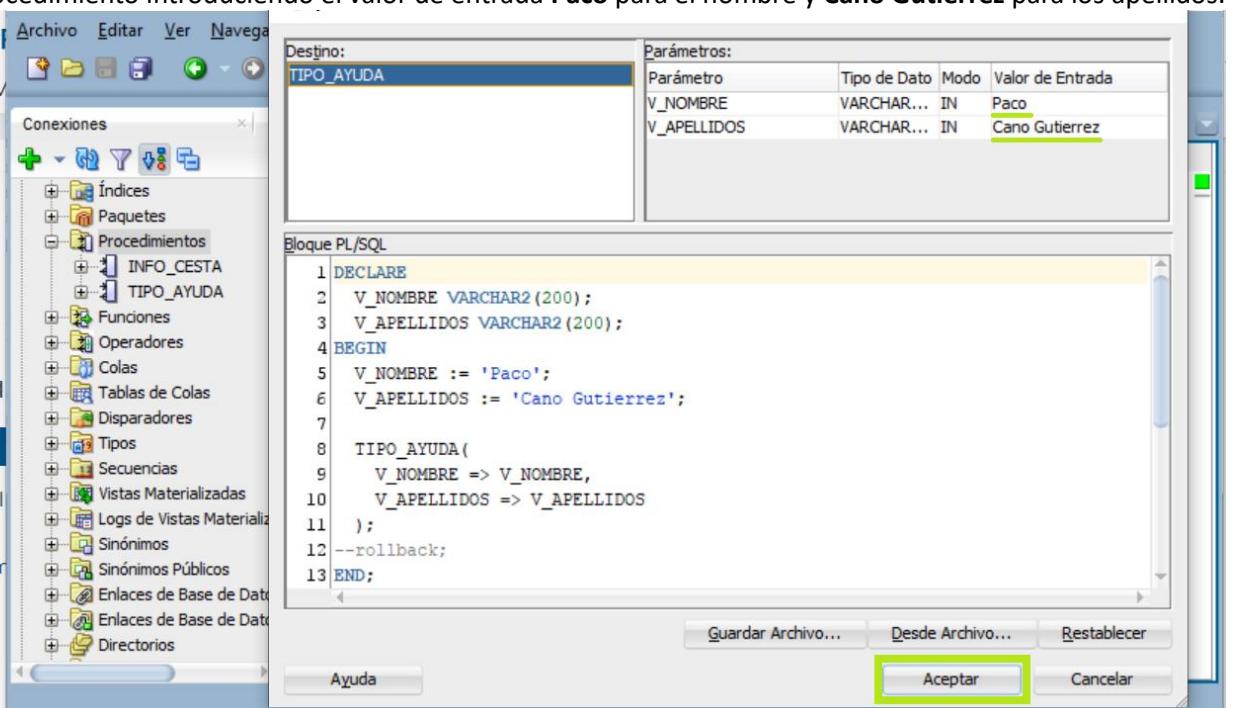
Compilamos el procedimiento **TIPO_AYUDA**.



-- Solamente si el tipo de avuda actual es diferente del tipo que había

Procedure TIPO_AYUDA compilado

Ejecutamos el procedimiento introduciendo el valor de entrada **Paco** para el nombre y **Cano Gutierrez** para los apellidos.



Parámetro	Tipo de Dato	Modo	Valor de Entrada
V_NOMBRE	VARCHAR...	IN	Paco
V_APELLIDOS	VARCHAR...	IN	Cano Gutierrez

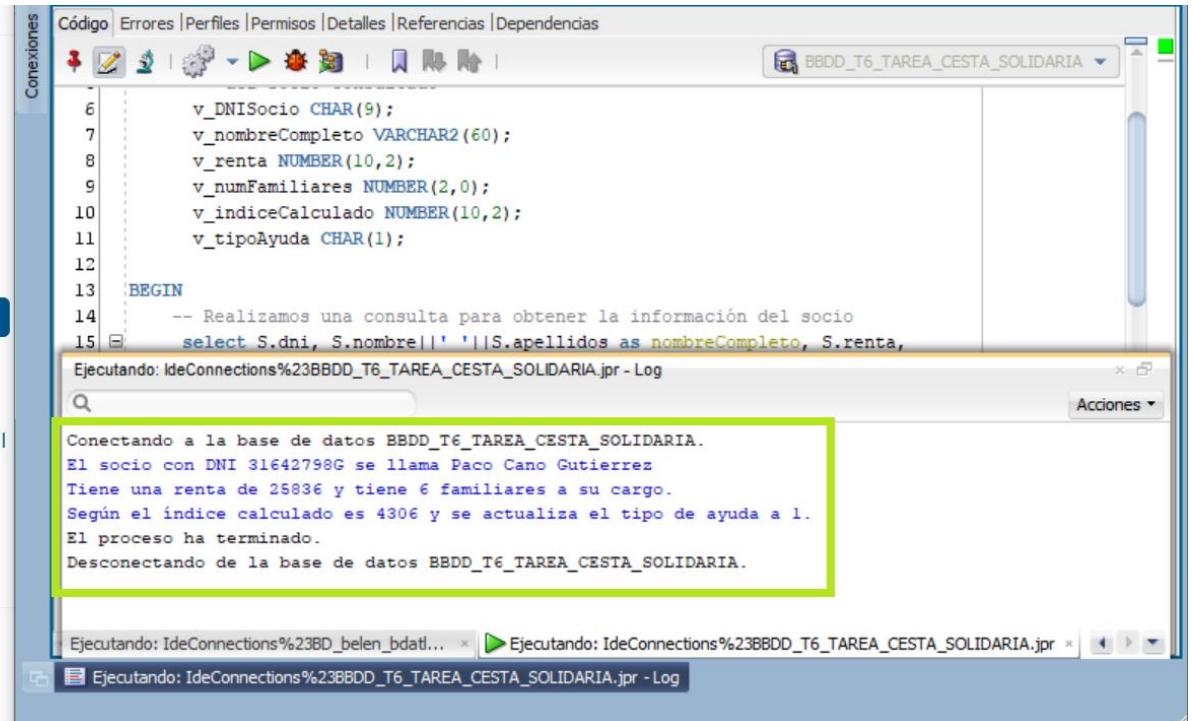
```

1 DECLARE
2   V_NOMBRE VARCHAR2(200);
3   V_APELLIDOS VARCHAR2(200);
4 BEGIN
5   V_NOMBRE := 'Paco';
6   V_APELLIDOS := 'Cano Gutierrez';
7
8   TIPO_AYUDA(
9     V_NOMBRE => V_NOMBRE,
10    V_APELLIDOS => V_APELLIDOS
11  );
12 --rollback;
13 END;

```

Al ejecutar el procedimiento podemos observar que se muestra por pantalla el DNI del socio, su nombre completo, la renta, el número de familiares a su cargo, el valor calculado respecto a la renta y número de familiares y por último el tipo de ayuda que recibirá.

El socio con DNI 31642798G se llama Paco Cano Gutierrez
 Tiene una renta de 25836 y tiene 6 familiares a su cargo.
 Según el índice calculado es 4306 y se actualiza el tipo de ayuda a 1.



The screenshot shows the Oracle SQL Developer interface. On the left, there's a sidebar with a profile picture of a woman and some personal information: "Solís Valle, Belén María" and a checked "Editar perfil" button. The main area has a toolbar at the top with various icons. Below the toolbar is a menu bar with "Código" selected. The code editor contains a PL/SQL block:

```

6      v_DNI_Socio CHAR(9);
7      v_nombreCompleto VARCHAR2(60);
8      v_renta NUMBER(10,2);
9      v_numFamiliares NUMBER(2,0);
10     v_indiceCalculado NUMBER(10,2);
11     v_tipoAyuda CHAR(1);
12
13    BEGIN
14        -- Realizamos una consulta para obtener la información del socio
15        select S.dni, S.nombre||' '||S.apellidos as nombreCompleto, S.renta,

```

The status bar at the bottom says "Ejecutando: IdeConnections%23BBDD_T6_TAREA_CESTA_SOLIDARIA.jpr - Log". A message window is open in the center, containing the output of the procedure execution:

```

Conectando a la base de datos BBDD_T6_TAREA_CESTA_SOLIDARIA.
El socio con DNI 31642798G se llama Paco Cano Gutierrez
Tiene una renta de 25836 y tiene 6 familiares a su cargo.
Según el indice calculado es 4306 y se actualiza el tipo de ayuda a 1.
El proceso ha terminado.
Desconectando de la base de datos BBDD_T6_TAREA_CESTA_SOLIDARIA.

```

Comprobamos que el procedimiento se ha ejecutado correctamente y se ha actualizado el tipo de ayuda correctamente. Como podemos apreciar ahora se le ha asignado a Paco Cano Gutierrez el tipo de ayuda 1, el cual coincide con el tipo de ayuda que se calculó.



The screenshot shows the Oracle SQL Developer interface again. The sidebar on the left has the same profile picture and personal information. The main area shows a table named "SOCIOS" with the following columns: DNI, NOMBRE, APELLIDOS, DIRECCION, TELEFONO, FECHA_N..., INGRESOS, NUMFAM, AYUDA, and RENTA. The table contains 13 rows of data. Row 1, corresponding to the executed procedure, has been highlighted in yellow. The "AYUDA" column for this row shows the value "1", which matches the calculated index of 4306.

DNI	NOMBRE	APELLIDOS	DIRECCION	TELEFONO	FECHA_N...	INGRESOS	NUMFAM	AYUDA	RENTA
1 31642798G	Paco	Cano Gutierrez	Calle Sabinar 8	617478552	31/01/87	600,5	6	1	25836
2 74819253B	Dario	Ferrer Martinez	Calle Alemania 11	678451947	01/05/00	350,6	4	(null)	18920
3 55487914P	Lara	Martin Rubio	Calle Mandarina 7	642772771	12/09/95	400,8	5	(null)	32589
4 27445115T	Alex	Perez Hernandez	Calle Segovia 8	643600501	04/12/99	604,25	6	(null)	25846
5 88544692S	Martirio	Batalha Salom	Calle Bilbao 2	666458123	31/01/57	100	3	(null)	25348
6 22951645V	Custodia	Guardiola Conesa	Avenida Juan Carlos 7	674184601	06/07/83	200	2	(null)	25820
7 27531479A	Isidro	Figueroas Perez	Calle Limon 73	700284012	26/11/77	100	3	(null)	18000
8 19311618X	Dimas	Cerdan Salinas	Calle Mares 14	792666320	29/03/89	300	3	(null)	60025
9 98452175H	Eduardo	Lopez Lopez	Avenida Gato Siames 26	620314785	23/05/96	1500	3	(null)	25000
10 57934526R	Reyes	Mayoral Valero	Calle Miguel Segura 12	724159043	27/11/88	2000	2	(null)	12546
11 26354517K	Antonia	Vicente Prat	Calle Maracas 22	658001200	28/11/77	1300	2	(null)	26358
12 33123987C	David	Pardo Marquez	Calle San Miguel 31	654004221	04/01/93	2000	4	(null)	15000
13 66225571H	Ulises	Garcia Sanz	Calle Fernando Alonso 33	647412349	18/10/86	1200	2	(null)	13258

The status bar at the bottom says "Ejecutando: IdeConnections%23BBDD_T6_TAREA_CESTA_SOLIDARIA.jpr - Log".

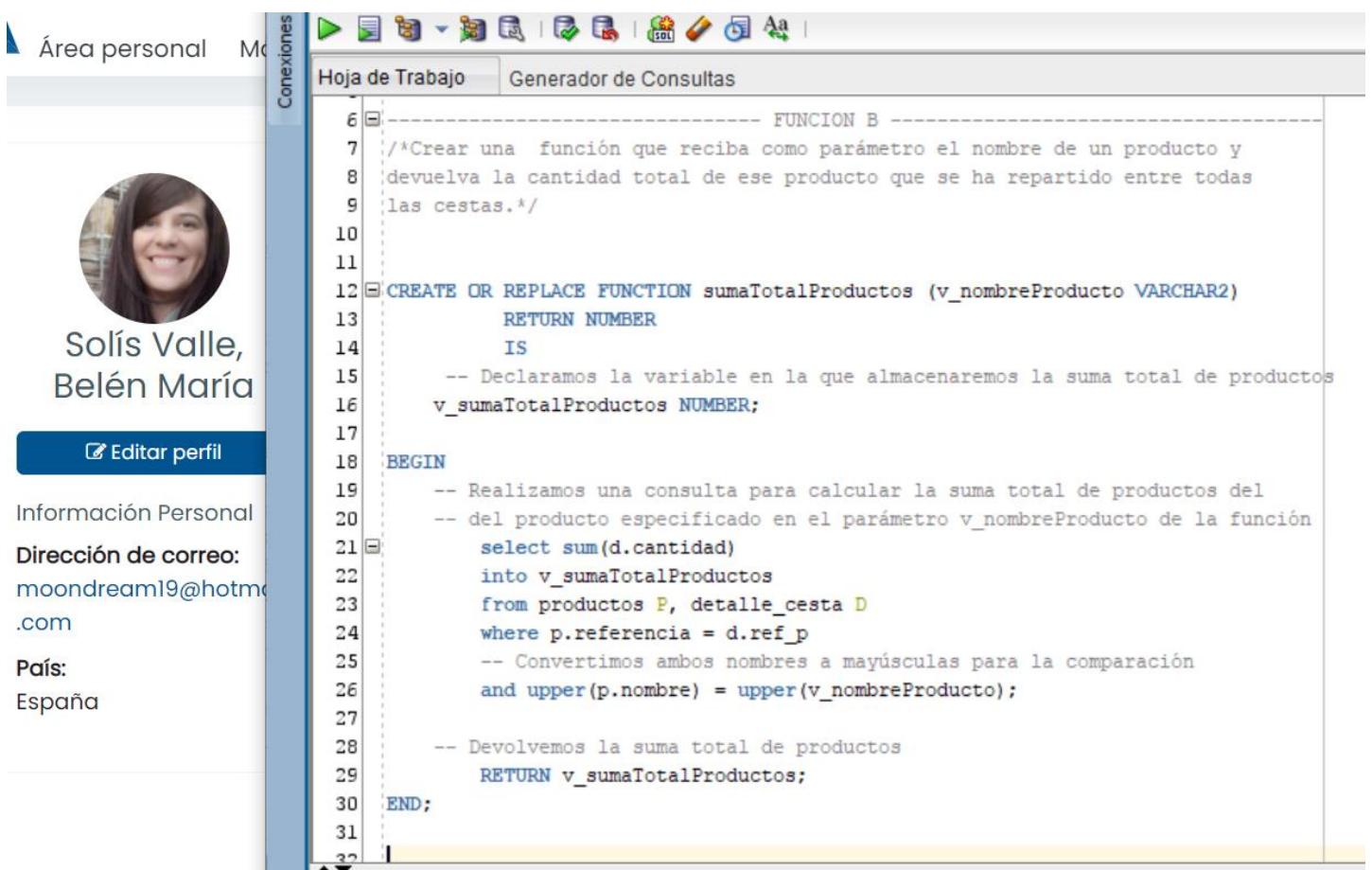
APARTADO C:

- C. Crear una función que reciba como parámetro el nombre de un producto y devuelva la cantidad total de ese producto que se ha repartido entre todas las cestas.

La función **sumaTotalProductos** toma un parámetro de entrada **v_nombreProducto** de tipo VARCHAR2, que es el nombre del producto del cual queremos obtener la cantidad total.

Declaramos la variable en la que almacenaremos la suma total de los productos: **v_sumaTotalProductos**. Tras lo cual, realizamos una consulta para calcular la suma total de productos del producto especificado. Esta consulta calcula la suma total de productos del producto especificado en todas las cestas.

El resultado de esa suma lo almacenamos en la variable **v_sumaTotalProductos**. Finalmente, la función devolverá el resultado de esta variable.



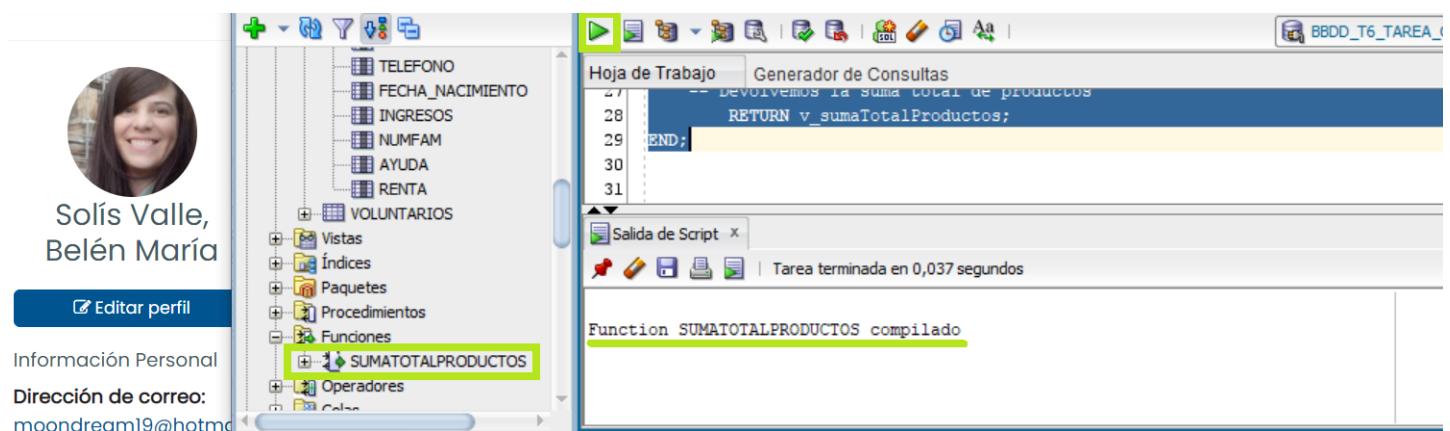
The screenshot shows the Oracle SQL Developer interface. On the left, there's a sidebar with a profile picture of a woman and user information: "Área personal" (Personal Area), "Solís Valle, Belén María", and "Editar perfil" (Edit Profile). Below that is "Información Personal" (Personal Information) with fields for "Dirección de correo:" (Email: moondream19@hotmail.com) and "País:" (Country: España). The main window is titled "Hoja de Trabajo" (Worksheet) and "Generador de Consultas" (Query Builder). It contains the following PL/SQL code:

```

6   -- FUNCION B
7   /*Crear una función que reciba como parámetro el nombre de un producto y
8   devuelva la cantidad total de ese producto que se ha repartido entre todas
9   las cestas.*/
10
11
12 CREATE OR REPLACE FUNCTION sumaTotalProductos (v_nombreProducto VARCHAR2)
13     RETURN NUMBER
14     IS
15         -- Declaramos la variable en la que almacenaremos la suma total de productos
16         v_sumaTotalProductos NUMBER;
17
18 BEGIN
19     -- Realizamos una consulta para calcular la suma total de productos del
20     -- del producto especificado en el parámetro v_nombreProducto de la función
21     select sum(d.cantidad)
22     into v_sumaTotalProductos
23     from productos P, detalle_cesta D
24     where p.referencia = d.ref_p
25     -- Convertimos ambos nombres a mayúsculas para la comparación
26     and upper(p.nombre) = upper(v_nombreProducto);
27
28     -- Devolvemos la suma total de productos
29     RETURN v_sumaTotalProductos;
30 END;
31
32

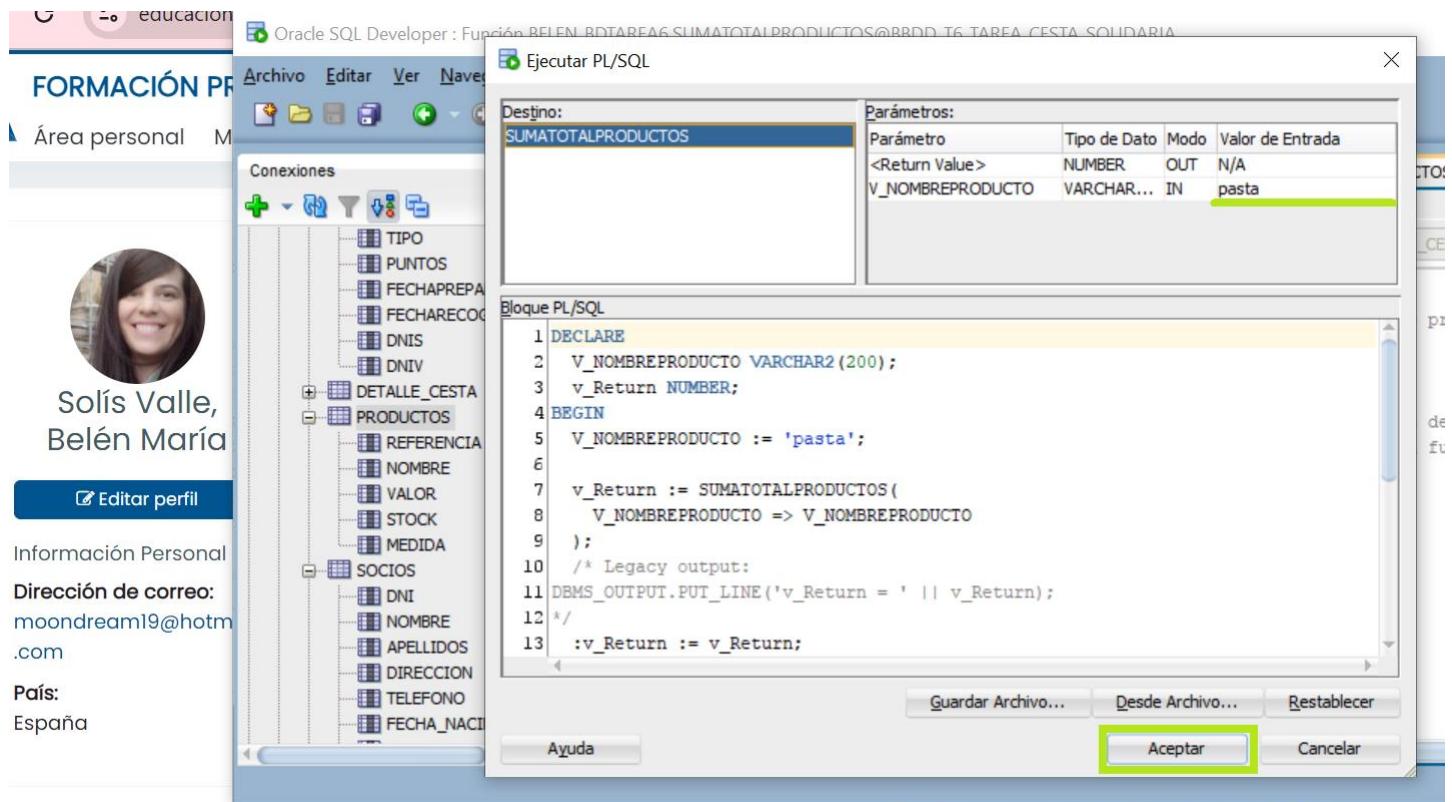
```

Compilamos la función **sumaTotalProductos**

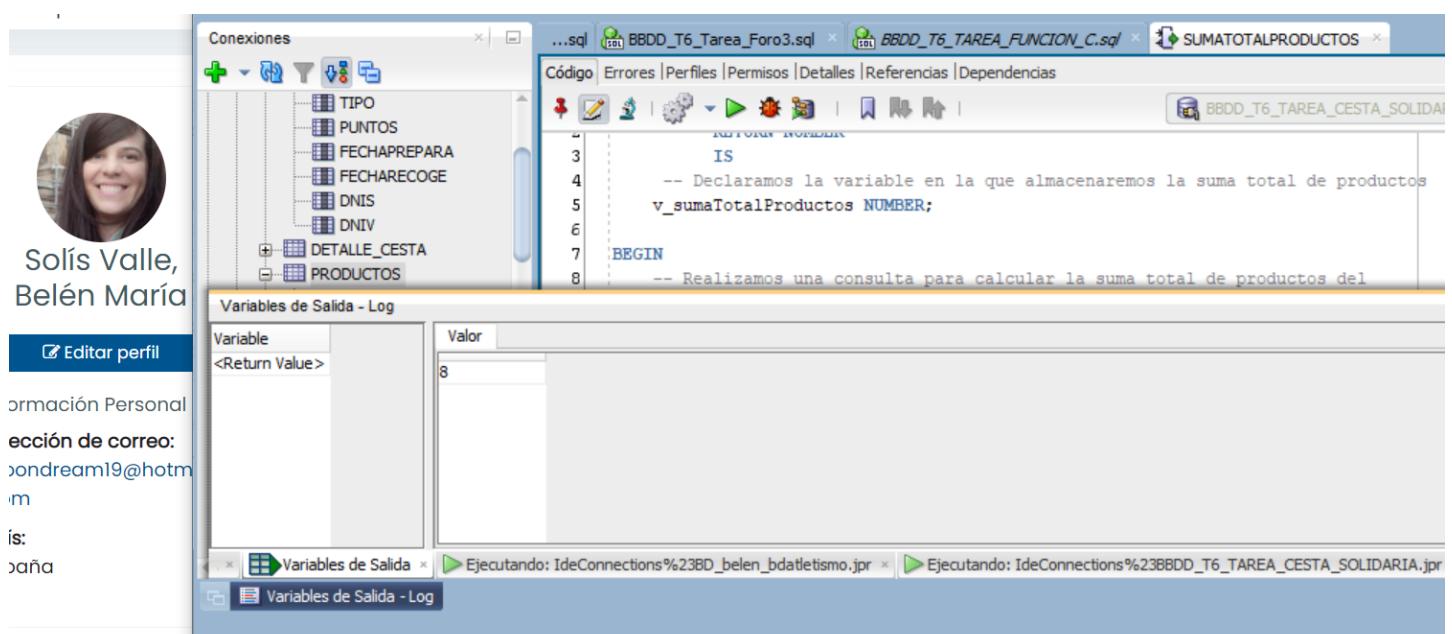


The screenshot shows the Oracle SQL Developer interface after compilation. The left sidebar now shows a tree structure with nodes like TELEFONO, FECHA_NACIMIENTO, INGRESOS, NUMFAM, AYUDA, RENTA, VOLUNTARIOS, Vistas, Índices, Paquetes, Procedimientos, and Funciones. The "Funciones" node has a child node named "SUMATOTALPRODUCTOS", which is highlighted with a green border. The main worksheet window shows the function definition again, and the status bar at the bottom right indicates "Tarea terminada en 0,037 segundos" (Task completed in 0,037 seconds). The output window below the worksheet shows the message "Function SUMATOTALPRODUCTOS compilado" (Function SUMATOTALPRODUCTOS compiled).

Ejecutamos la función estableciendo como valor de entrada **pasta**, tras lo cual aceptamos.



Tras la ejecución, nos dirigimos a la pestaña **Variables de Salida**, como resultado de salida nos devuelve el valor **8**. He consultado las líneas de pedido y efectivamente el valor corresponde con el total del producto de ese producto en todas las cestas.

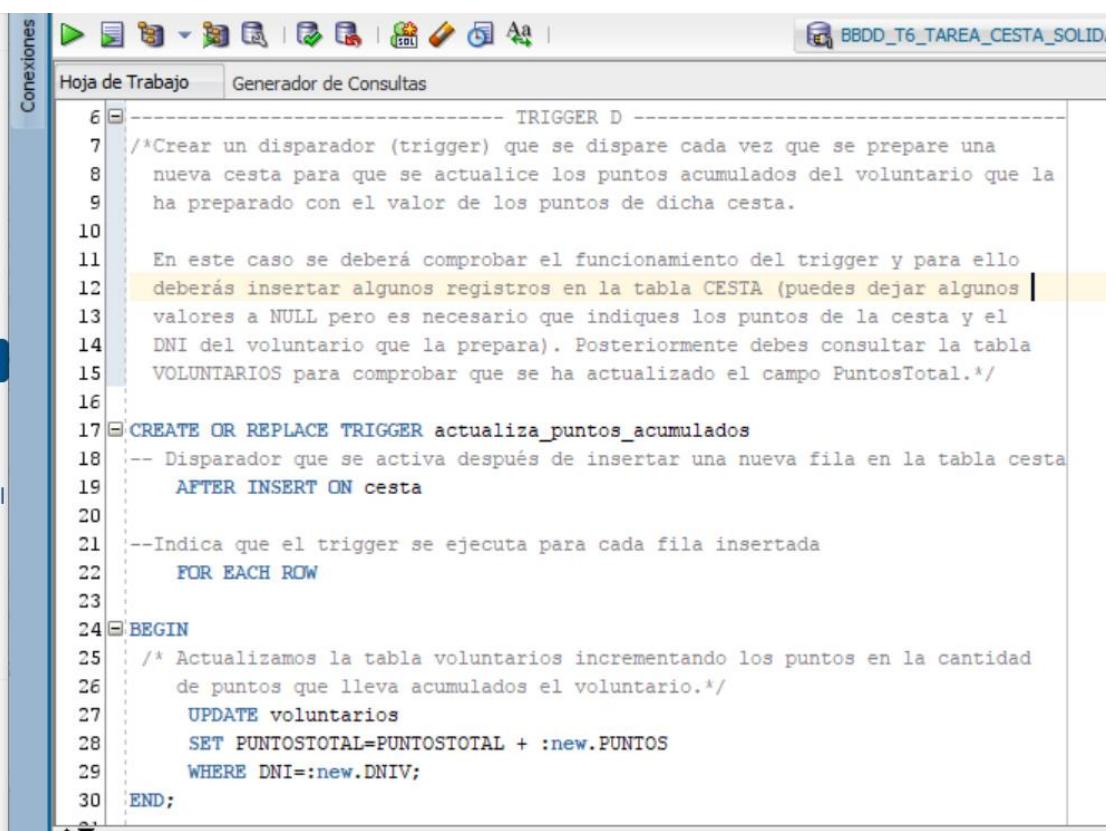


APARTADO D:

D. Crear un disparador (trigger) que se dispare cada vez que se prepare una nueva cesta para que se actualice los puntos acumulados del voluntario que la ha preparado con el valor de los puntos de dicha cesta.

- En este caso se deberá comprobar el funcionamiento del trigger y para ello deberás insertar algunos registros en la tabla CESTA (puedes dejar algunos valores a NULL, pero es necesario que indiques los puntos de la cesta y el DNI del voluntario que la prepara). Posteriormente debes consultar la tabla VOLUNTARIOS para comprobar que se ha actualizado el campo PuntosTotal.

Este disparador/trigger **se activa automáticamente después de que se inserta una nueva fila en la tabla cesta**. Su función es **actualizar los puntos acumulados del voluntario** que ha preparado la cesta con el valor de los puntos de dicha cesta. Para lograr esto, suma los puntos de la nueva cesta al total de puntos acumulados del voluntario correspondiente en la tabla voluntarios.

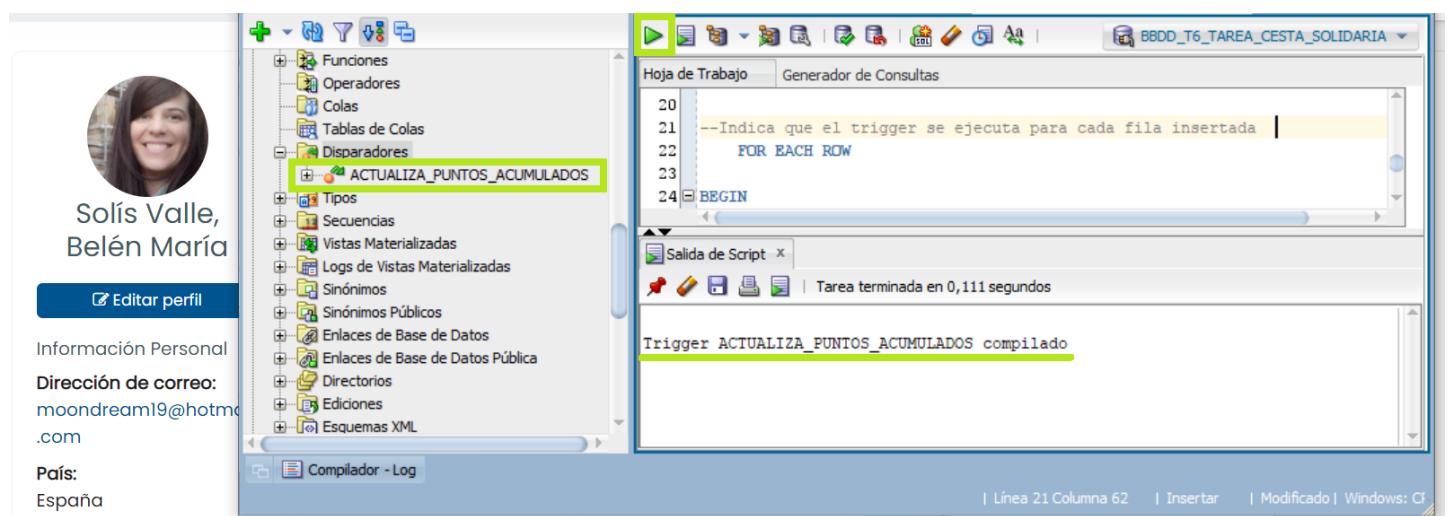


```

TRIGGER D
/*
  Crear un disparador (trigger) que se dispare cada vez que se prepare una
  nueva cesta para que se actualice los puntos acumulados del voluntario que la
  ha preparado con el valor de los puntos de dicha cesta.
*/
En este caso se deberá comprobar el funcionamiento del trigger y para ello
deberás insertar algunos registros en la tabla CESTA (puedes dejar algunos |
valores a NULL pero es necesario que indiques los puntos de la cesta y el
DNI del voluntario que la prepara). Posteriormente debes consultar la tabla
VOLUNTARIOS para comprobar que se ha actualizado el campo PuntosTotal.*/
CREATE OR REPLACE TRIGGER actualiza_puntos_acumulados
-- Disparador que se activa después de insertar una nueva fila en la tabla cesta
  AFTER INSERT ON cesta
--Indica que el trigger se ejecuta para cada fila insertada
  FOR EACH ROW
BEGIN
  /* Actualizamos la tabla voluntarios incrementando los puntos en la cantidad
  de puntos que lleva acumulados el voluntario.*/
  UPDATE voluntarios
  SET PUNTOSTOTAL=PUNTOSTOTAL + :new.PUNTOS
  WHERE DNI=:new.DNIV;
END;

```

Compilamos el trigger **actualiza_puntos_acumulados**





Solís Valle,
Belén María

Editar perfil

Información Personal

Dirección de correo:
moondream19@hotmail.com

País:
España

Hoja de Trabajo Generador de Consultas BBDD_T6_TAREA_CESTA_SOLIDARIA

```

20
21 -- Indica que el trigger se ejecuta para cada fila insertada |
22   FOR EACH ROW
23
24 BEGIN

```

Salida de Script

Tarea terminada en 0,111 segundos

Trigger ACTUALIZA_PUNTOS_ACUMULADOS compilado

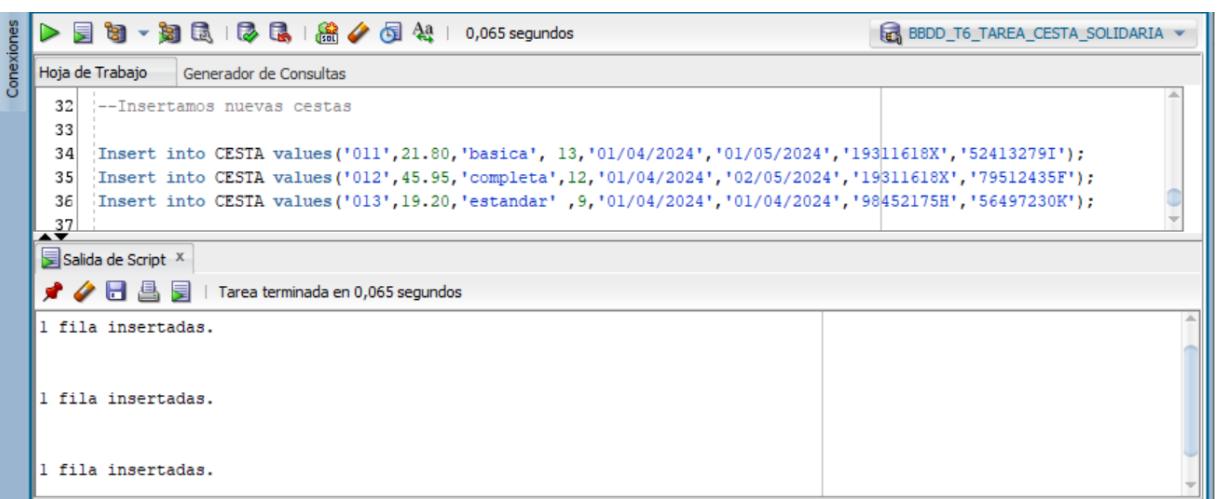
Para comprobar el buen funcionamiento de este trigger comenzamos consultando la tabla voluntarios y comprobamos los puntos que tiene cada voluntario antes de insertar nuevas cestas



Conexiones Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL Ordenar... Filtrar: Acciones...

DNI	NOMBRE	APELLIDOS	DIRECCION	TELEFONO	FECHA...	PUNTOSTOTAL
1 79512435F	Juan	Moreno Fernandez	Calle Salamanca 77	621325079	30/10/74	100
2 55976143M	Maria Isabel	Beltran Lopez	Calle Perro 56	691023456	05/12/79	200
3 40139191X	Francisca	Becerra Zapata	Calle de la Rosa 12	667884221	13/12/92	200
4 77421335E	David	Gonzalez Azcona	Calle Madrid 17	634008222	25/10/98	400
5 56497230K	Jose Luis	Arce Navarro	Calle Berlin 12	742851002	28/02/86	500
6 52413279I	Amaia	Valverde Lopez	Avenida Marco Polo 32	657035042	13/09/90	500
7 10942404Y	Cosme	Nogueira Rebollo	Calle del Motor 33	605203665	16/06/73	650
8 91471235B	Jesus	Cruz Cobo	Avenida Calvo Sotelo 27	649651036	27/04/95	800
9 27555115T	Alejandra	Collado Hernandez	Calle Solaz 9	643600701	04/11/99	1001
10 22447896V	Patricia	Fernandez Ruiz	Calle Japon 1	624752004	14/07/97	1500

Insertamos nuevas cestas en la tabla cesta



Conexiones Hoja de Trabajo Generador de Consultas BBDD_T6_TAREA_CESTA_SOLIDARIA

```

32 --Insertamos nuevas cestas
33
34 Insert into CESTA values('011',21.80,'basica', 13,'01/04/2024','01/05/2024','19311618X','52413279I');
35 Insert into CESTA values('012',45.95,'completa',12,'01/04/2024','02/05/2024','19311618X','79512435F');
36 Insert into CESTA values('013',19.20,'estandar' ,9,'01/04/2024','01/04/2024','98452175H','56497230K');
37

```

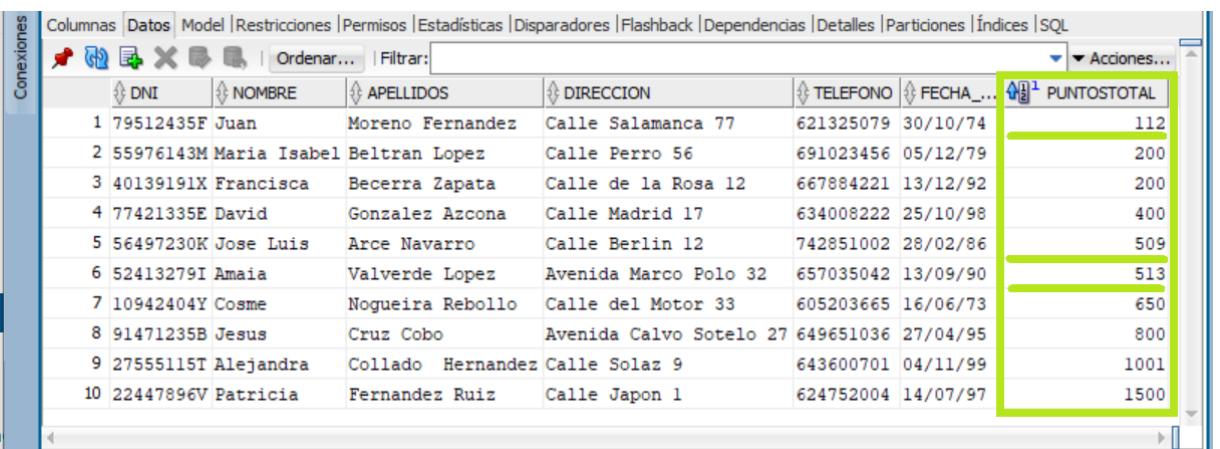
Salida de Script Tarea terminada en 0,065 segundos

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

Comprobamos si se han actualizado los puntos obtenidos por los voluntarios, como podemos apreciar los puntos totales de los voluntarios se han actualizado correctamente.



Conexiones Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL Ordenar... Filtrar: Acciones...

DNI	NOMBRE	APELLIDOS	DIRECCION	TELEFONO	FECHA...	PUNTOSTOTAL
1 79512435F	Juan	Moreno Fernandez	Calle Salamanca 77	621325079	30/10/74	112
2 55976143M	Maria Isabel	Beltran Lopez	Calle Perro 56	691023456	05/12/79	200
3 40139191X	Francisca	Becerra Zapata	Calle de la Rosa 12	667884221	13/12/92	200
4 77421335E	David	Gonzalez Azcona	Calle Madrid 17	634008222	25/10/98	400
5 56497230K	Jose Luis	Arce Navarro	Calle Berlin 12	742851002	28/02/86	509
6 52413279I	Amaia	Valverde Lopez	Avenida Marco Polo 32	657035042	13/09/90	513
7 10942404Y	Cosme	Nogueira Rebollo	Calle del Motor 33	605203665	16/06/73	650
8 91471235B	Jesus	Cruz Cobo	Avenida Calvo Sotelo 27	649651036	27/04/95	800
9 27555115T	Alejandra	Collado Hernandez	Calle Solaz 9	643600701	04/11/99	1001
10 22447896V	Patricia	Fernandez Ruiz	Calle Japon 1	624752004	14/07/97	1500