

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

12-5-2020

Bases de datos objeto-relacionales.

Unidad 7.

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

David López Coronel.
Base de Datos. Grupo(A).

1. a) Crea el tipo de objeto **“Usuario”** con los siguientes atributos:

DNI VARCHAR2(9),
nombre VARCHAR2(30),
apellidos VARCHAR2(40),
domicilio VARCHAR2(50),
fNacimiento DATE

b) Crea, como tipo heredado de **“Usuario”**, el tipo de objeto **“Tecnico”** con los siguientes atributos:

salario NUMBER(6,2),
puntos_acumulados NUMBER(4),
incentivo NUMBER(4)

c) Crea el tipo de objeto **“Incidencia”** con los siguientes atributos:

ID VARCHAR2(3),
descripcion VARCHAR2(60),
estado VARCHAR2(20),
puntos_coste NUMBER(4),
f_abierta DATE,
f_repara DATE,
reftecnico REF Tecnico

d) Crea, como tipo heredado de **“Usuario”**, el tipo de objeto **“Cliente”** con los siguientes atributos:

IncidenciaU Incidencia

--a)

CREATE OR REPLACE TYPE *Usuario* AS OBJECT (

DNI VARCHAR2(9),

nombre VARCHAR2(30),

apellidos VARCHAR2(40),

domicilio VARCHAR2(50),

fNacimiento DATE

)NOT FINAL;

/

--b)

CREATE OR REPLACE TYPE *Tecnico* UNDER *Usuario* (

salario NUMBER(6,2),

puntos_acumulados NUMBER(4),

incentivo NUMBER(4)

);

/

--c)

CREATE OR REPLACE TYPE *Incidencia* AS OBJECT (

ID VARCHAR2(3),

descripcion VARCHAR2(60),

estado VARCHAR2(20),

puntos_coste NUMBER(4),

f_abierta DATE,

f_repara DATE,

reftecnico REF *Tecnico*

);

--d)

CREATE OR REPLACE TYPE *Cliente* UNDER *Usuario* (

IncidenciaU *Incidencia*

);

Capturas de pantalla Actividad 1: apartados a, b, d, c.

This screenshot shows the SQL Developer interface with the 'Tarea_Uni7' task selected. The left sidebar displays the database schema, with the 'Tipos' (Types) folder highlighted. The main editor window shows the following SQL script:

```
--a)
CREATE OR REPLACE TYPE Usuario AS OBJECT (
  DNI VARCHAR2(9),
  nombre VARCHAR2(30),
  apellidos VARCHAR2(40),
  domicilio VARCHAR2(50),
  fNacimiento DATE
) NOT FINAL;
/
--b)
CREATE OR REPLACE TYPE Tecnico UNDER Usuario (
  salario NUMBER(6,2),
  puntos_acumulados NUMBER(4),
  incentivo NUMBER(4)
);
/
CREATE OR REPLACE TYPE Cliente UNDER Usuario (
  Incidencia INCIDENCIA
);
/
```

The 'Salida de Script' (Script Output) window below shows the compilation results:

```
Type USUARIO compilado
Type TECNICO compilado
Type CLIENTE compilado
```

The right sidebar shows the user profile for David López Coronel, and the system clock indicates 20:55 on 12/05/2020.

This screenshot shows the SQL Developer interface with the 'Tarea_Uni7' task selected. The left sidebar displays the database schema, with the 'Tipos' (Types) folder highlighted. The main editor window shows the following SQL script:

```
--c)
CREATE OR REPLACE TYPE Incidencia AS OBJECT (
  ID VARCHAR2(3),
  descripcion VARCHAR2(60),
  estado VARCHAR2(20),
  puntos_coste NUMBER(4),
  f_abierta DATE,
  f_repara DATE,
  ref tecnico REF Tecnico
);
/
```

The 'Salida de Script' (Script Output) window below shows the compilation result:

```
Type INCIDENCIA compilado
```

The right sidebar shows the user profile for David López Coronel, and the system clock indicates 20:52 on 12/05/2020.

2. Crea un método constructor para el tipo de objeto “**Tecnico**”, en el que se indiquen como parámetros el DNI, nombre, apellidos, domicilio, fNacimiento, salario y puntos_acumulados. Este método debe de calcular y asignar el valor del atributo "incentivo" que será de 10 euros por cada punto_acumulado.

```
CREATE OR REPLACE TYPE Tecnico UNDER Usuario (
```

```
    salario NUMBER(6,2),
```

```
    puntos_acumulados NUMBER(4),
```

```
    incentivo NUMBER(4),
```

```
    CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2,  
    apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER,  
    puntos_acumulados NUMBER)
```

```
    RETURN SELF AS RESULT);
```

```
CREATE OR REPLACE TYPE BODY Tecnico AS
```

```
    CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2,  
    apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER,  
    puntos_acumulados NUMBER)
```

```
    RETURN SELF AS RESULT
```

```
    IS
```

```
        BEGIN
```

```
            SELF.DNI:=DNI;
```

```
            SELF.nombre:=nombre;
```

```
            SELF.apellidos:=apellidos;
```

```
            SELF.domicilio:=domicilio;
```

```
            SELF.fNacimiento:=fNacimiento;
```

```
            SELF.salario:=salario;
```

```
            SELF.puntos_acumulados:=puntos_acumulados;
```

```
            SELF.incentivo:=10 * puntos_acumulados;
```

```
            RETURN;
```

```
        END;
```

```
END;
```

--Insertare este método o procedimiento para comprobar mi método constructor realizado en esta actividad 2), el cual nos calcule automáticamente el incentivo:

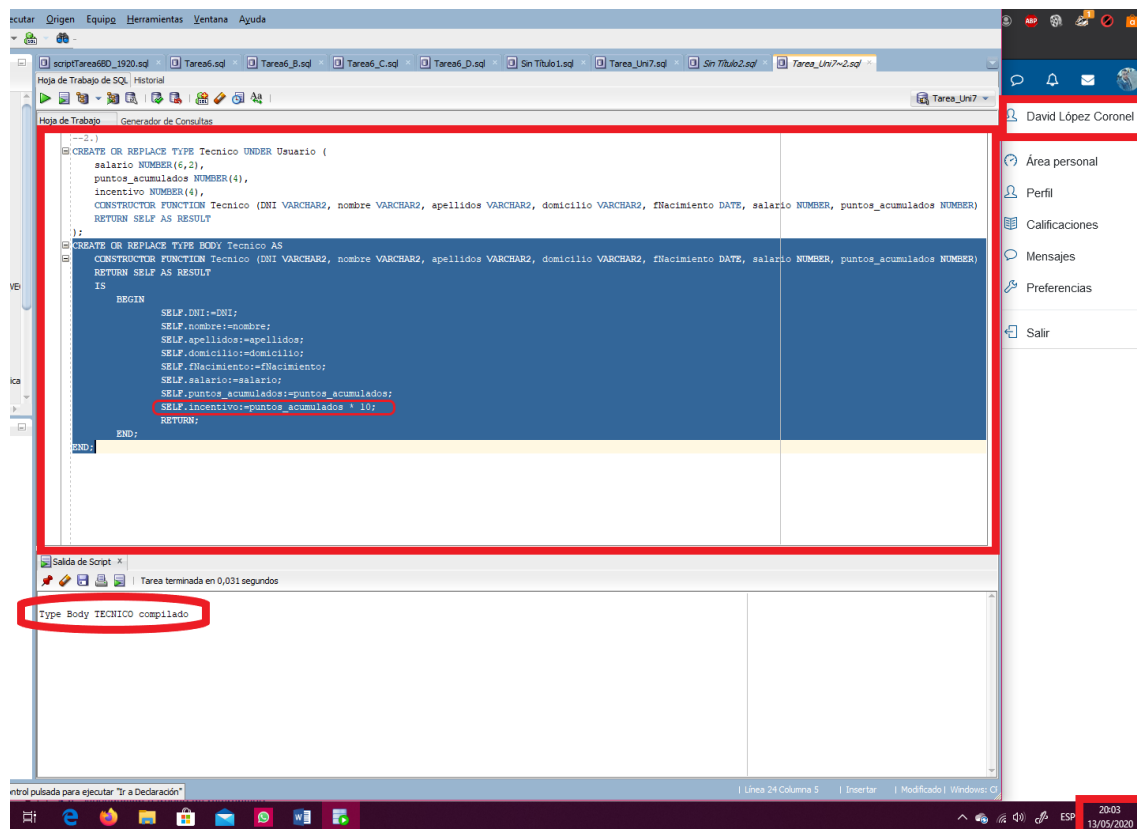
```
tIncidencia003 := NEW tecnico('39556378C', 'David', 'Lopez Coronel',  
'domicilio', '22-01-1991', 1600, 350);
```

```
dbms_output.put_line('El salario, puntos acumulados e incentivo  
de la Incidencia 002 es:
```

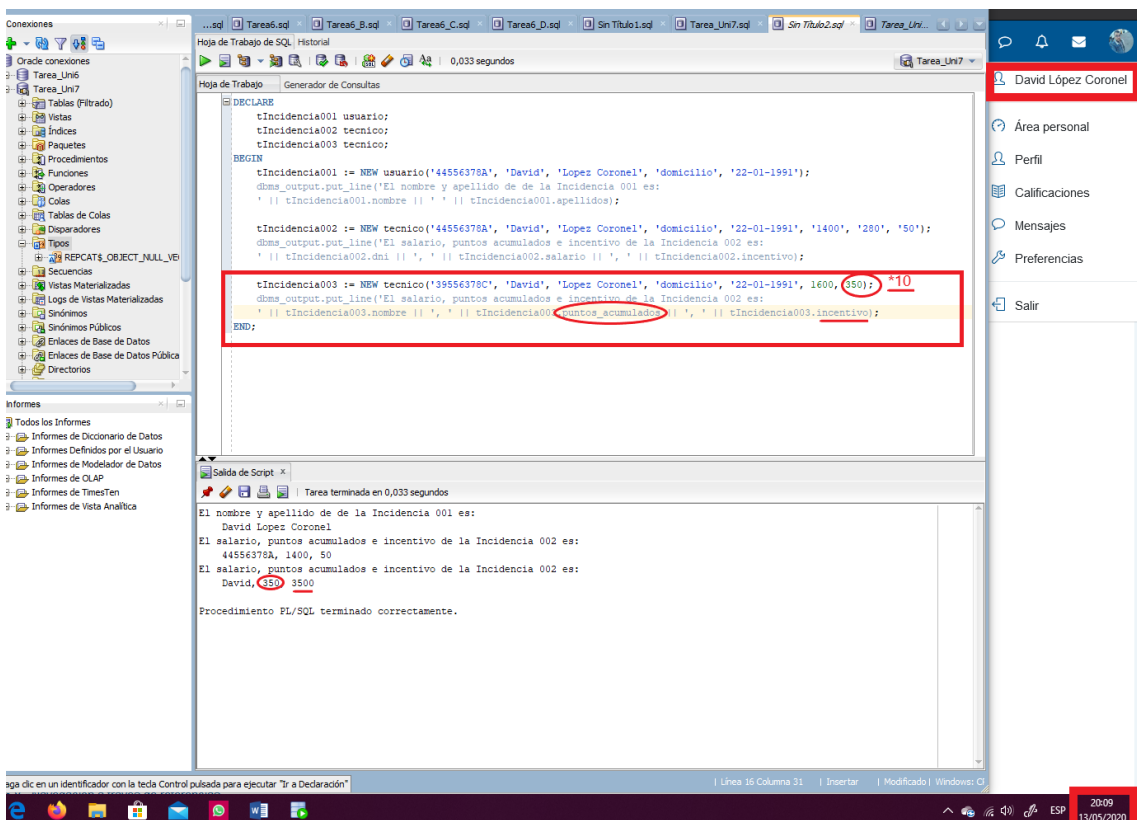
```
' || tIncidencia003.nombre || ', ' || tIncidencia003.puntos_acumulados ||  
'; ' || tIncidencia003.incentivo);
```

Estas son las capturas de la actividad 2); el método constructor

‘incentivo’= puntos_acumulados * 10



Aquí comprobamos y realizamos el procedimiento para ver como calcula automáticamente sin insertar el incentivo:

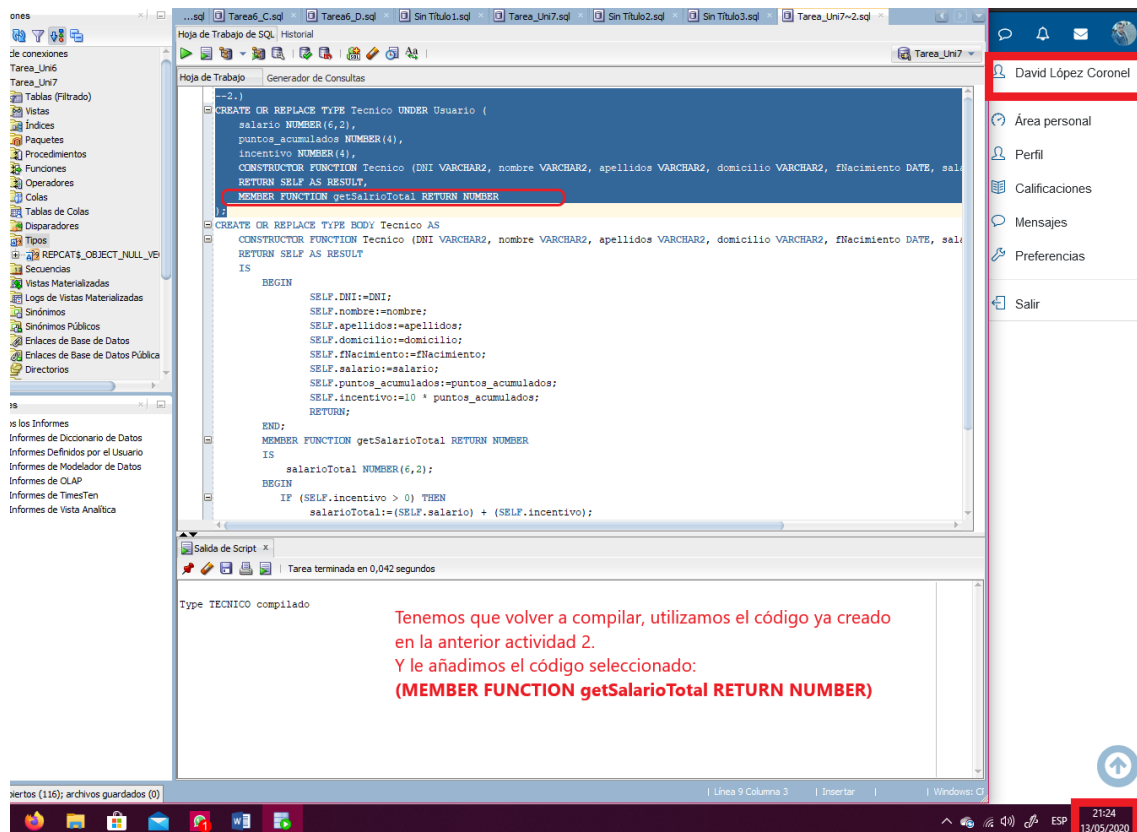


3. Crea un método "**getSalarioTotal**" para el tipo de objetos "**Tecnico**" que permita obtener el salario total como la suma de los campos: salario + incentivo.

```
CREATE OR REPLACE TYPE Tecnico UNDER Usuario (  
    salario NUMBER(6,2),  
    puntos_acumulados NUMBER(4),  
    incentivo NUMBER(4),  
    CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2,  
    apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER,  
    puntos_acumulados NUMBER)  
    RETURN SELF AS RESULT,  
    MEMBER FUNCTION getSalarioTotal RETURN NUMBER  
);  
  
CREATE OR REPLACE TYPE BODY Tecnico AS  
    CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2,  
    apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER,  
    puntos_acumulados NUMBER)  
    RETURN SELF AS RESULT  
    IS  
        BEGIN  
            SELF.DNI:=DNI;  
            SELF.nombre:=nombre;  
            SELF.apellidos:=apellidos;  
            SELF.domicilio:=domicilio;  
            SELF.fNacimiento:=fNacimiento;  
            SELF.salario:=salario;  
            SELF.puntos_acumulados:=puntos_acumulados;  
            SELF.incentivo:=10 * puntos_acumulados;  
            RETURN;  
        END;
```

```
MEMBER FUNCTION getSalarioTotal RETURN NUMBER
IS
    salarioTotal NUMBER(6,2);
BEGIN
    IF (SELF.incentivo > 0) THEN
        salarioTotal := (SELF.salario) + (SELF.incentivo);
    ELSE salarioTotal := (SELF.salario);
    END IF;
    RETURN(salarioTotal);
END getSalarioTotal;
END;
```

Añado las capturas de esta actividad 3:



Hoja de Trabajo de SQL: Historial

Generador de Consultas

```
--2.)
CREATE OR REPLACE TYPE Tecnico UNDER Usuario (
  salario NUMBER(6,2),
  puntos_acumulados NUMBER(4),
  incentivo NUMBER(4),
  CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2, apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER, puntos_acumulados NUMBER, incentivo NUMBER)
  RETURN SELF AS RESULT,
  MEMBER FUNCTION getSalarioTotal RETURN NUMBER
)
CREATE OR REPLACE TYPE BODY Tecnico AS
  CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2, apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER, puntos_acumulados NUMBER, incentivo NUMBER)
  RETURN SELF AS RESULT
  IS
  BEGIN
    SELF.DNI:=DNI;
    SELF.nombre:=nombre;
    SELF.apellidos:=apellidos;
    SELF.domicilio:=domicilio;
    SELF.fNacimiento:=fNacimiento;
    SELF.salario:=salario;
    SELF.puntos_acumulados:=puntos_acumulados;
    SELF.incentivo:=10 * puntos_acumulados;
    RETURN;
  END;
  MEMBER FUNCTION getSalarioTotal RETURN NUMBER
  IS
    salarioTotal NUMBER(6,2);
  BEGIN
    IF (SELF.incentivo > 0) THEN
      salarioTotal:=(SELF.salario) + (SELF.incentivo);
    ELSE
      salarioTotal:=SELF.salario;
    END IF;
    RETURN (salarioTotal);
  END getSalarioTotal;
END;
```

Salida de Script x

Tarea terminada en 0,042 segundos

Type TECNICO compilado

Tenemos que volver a compilar, utilizamos el código ya creado en la anterior actividad 2.
Y le añadimos el código seleccionado:
(MEMBER FUNCTION getSalarioTotal RETURN NUMBER)

David López Coronel

Área personal

Perfil

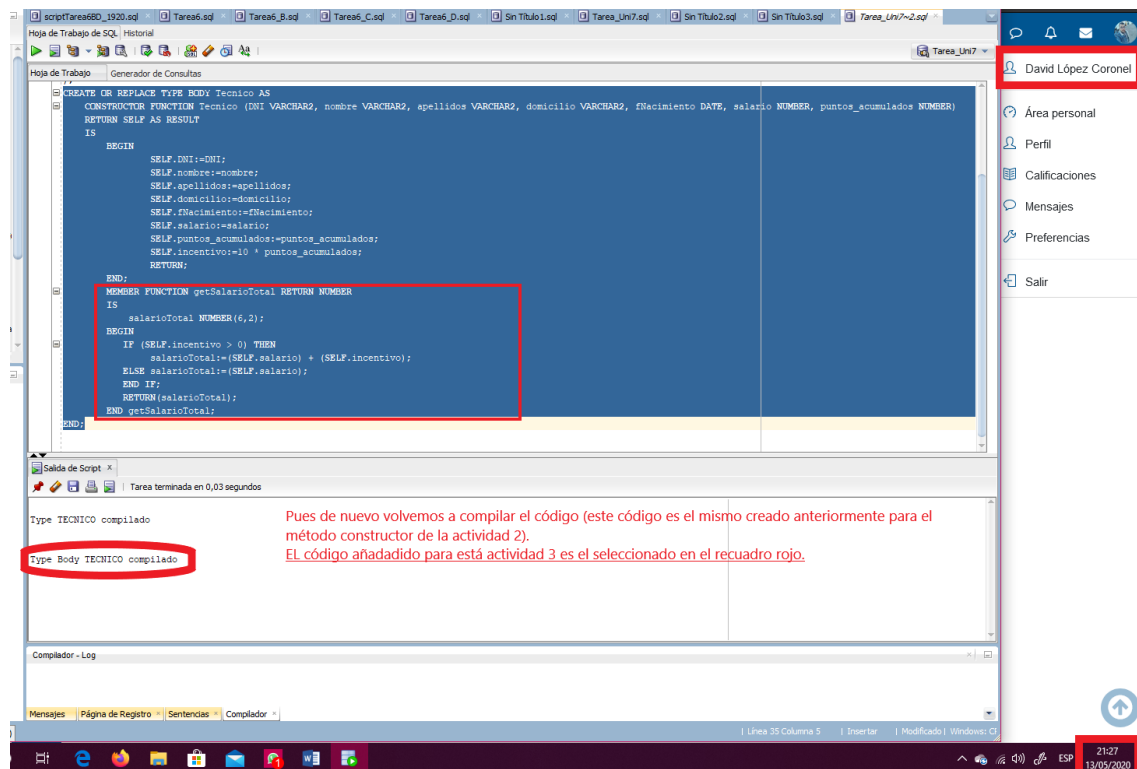
Calificaciones

Mensajes

Preferencias

Salir

21:24
13/05/2020



Hoja de Trabajo de SQL: Historial

Generador de Consultas

```
--2.)
CREATE OR REPLACE TYPE TECNICO AS
  CONSTRUCTOR FUNCTION Tecnico (DNI VARCHAR2, nombre VARCHAR2, apellidos VARCHAR2, domicilio VARCHAR2, fNacimiento DATE, salario NUMBER, puntos_acumulados NUMBER)
  RETURN SELF AS RESULT
  IS
  BEGIN
    SELF.DNI:=DNI;
    SELF.nombre:=nombre;
    SELF.apellidos:=apellidos;
    SELF.domicilio:=domicilio;
    SELF.fNacimiento:=fNacimiento;
    SELF.salario:=salario;
    SELF.puntos_acumulados:=puntos_acumulados;
    SELF.incentivo:=10 * puntos_acumulados;
    RETURN;
  END;
  MEMBER FUNCTION getSalarioTotal RETURN NUMBER
  IS
    salarioTotal NUMBER(6,2);
  BEGIN
    IF (SELF.incentivo > 0) THEN
      salarioTotal:=(SELF.salario) + (SELF.incentivo);
    ELSE
      salarioTotal:=SELF.salario;
    END IF;
    RETURN (salarioTotal);
  END getSalarioTotal;
END;
```

Salida de Script x

Tarea terminada en 0,03 segundos

Type TECNICO compilado

Type Body TECNICO compilado

Pues de nuevo volvemos a compilar el código (este código es el mismo creado anteriormente para el método constructor de la actividad 2).
El código añadido para esta actividad 3 es el seleccionado en el recuadro rojo.

David López Coronel

Área personal

Perfil

Calificaciones

Mensajes

Preferencias

Salir

21:27
13/05/2020

Y por último esta captura de la actividad 3, voy a comprobar el método “getSalarioTotal”

Para ello insertare el siguiente procedimiento :

```
dbms_output.put_line('El salario Total de la incidencia 003 es =' ||  
tIncidencia003.getSalarioTotal());
```

El cual da como resultado 5100. (Esto se debe a que tenemos un salario de 1600 más un incentivo de 3500)

The screenshot displays the SQL Developer interface. The main window shows a PL/SQL procedure named 'PROCEDIMIENTO PL/SQL terminado correctamente.' The procedure logic is as follows:

```
DECLARE  
tIncidencia001 usuario;  
tIncidencia002 tecnico;  
tIncidencia003 tecnico;  
  
BEGIN  
tIncidencia001 := NEW usuario('44556378A', 'David', 'Lopez Coronel', 'domicilio', '22-01-1991');  
dbms_output.put_line('El nombre y apellido de la Incidencia 001 es:  
' || tIncidencia001.nombre || ' ' || tIncidencia001.apellidos);  
  
tIncidencia002 := NEW tecnico('44556378A', 'David', 'Lopez Coronel', 'domicilio', '22-01-1991', '1400', '280', '50');  
dbms_output.put_line('El DNI, salario e incentivo de la Incidencia 002 es:  
' || tIncidencia002.dni || ' ' || tIncidencia002.salario || ' ' || tIncidencia002.incentivo);  
  
tIncidencia003 := NEW tecnico('39556378C', 'David', 'Lopez Coronel', 'domicilio', '22-01-1991', 1600, 3500);  
dbms_output.put_line('El nombre, puntos acumulados e incentivo calculado automaticamente a partir de (puntos_acumulados) *10 de la Incidencia 002 es:  
' || tIncidencia003.nombre || ' ' || tIncidencia003.puntos_acumulados || ' ' || tIncidencia003.incentivo);  
  
dbms_output.put_line('El salario Total de la incidencia 003 es =' || tIncidencia003.getSalarioTotal());  
END;
```

The 'Salida de Script' window shows the execution results:

```
El nombre y apellido de la Incidencia 001 es:  
David Lopez Coronel  
El DNI, salario e incentivo de la Incidencia 002 es:  
44556378A, 1400, 50  
El nombre, puntos acumulados e incentivo calculado automaticamente a partir de (puntos_acumulados) *10 de la Incidencia 002 es:  
David, 350, 3500  
El salario Total de la incidencia 003 es =5100  
Procedimiento PL/SQL terminado correctamente.
```

The 'Compilador - Log' window shows the compilation status: 'Procedimiento PL/SQL terminado correctamente.'

4. Crea una tabla "**Tecnicos**" de objetos de tipo "**Tecnico**". Inserta en dicha tabla dos objetos de tipo "**Tecnico**".

```
CREATE TABLE tecnicos OF tecnico;
```

```
INSERT INTO tecnicos VALUES(tecnico ('47589625J', 'Manuel', 'Sánchez  
Torra', 'C/ Roma, 42 Sevilla', '06/10/1971', 1800, 50, 500));
```

```
DECLARE
```

```
    tIncidencia003 tecnico;
```

```
BEGIN
```

```
    tIncidencia003 := NEW tecnico('48626742K', 'Carmen', 'Diaz Perez', 'C/  
Leonardo 26, Huelva', '10/06/1987', 1200, 30);
```

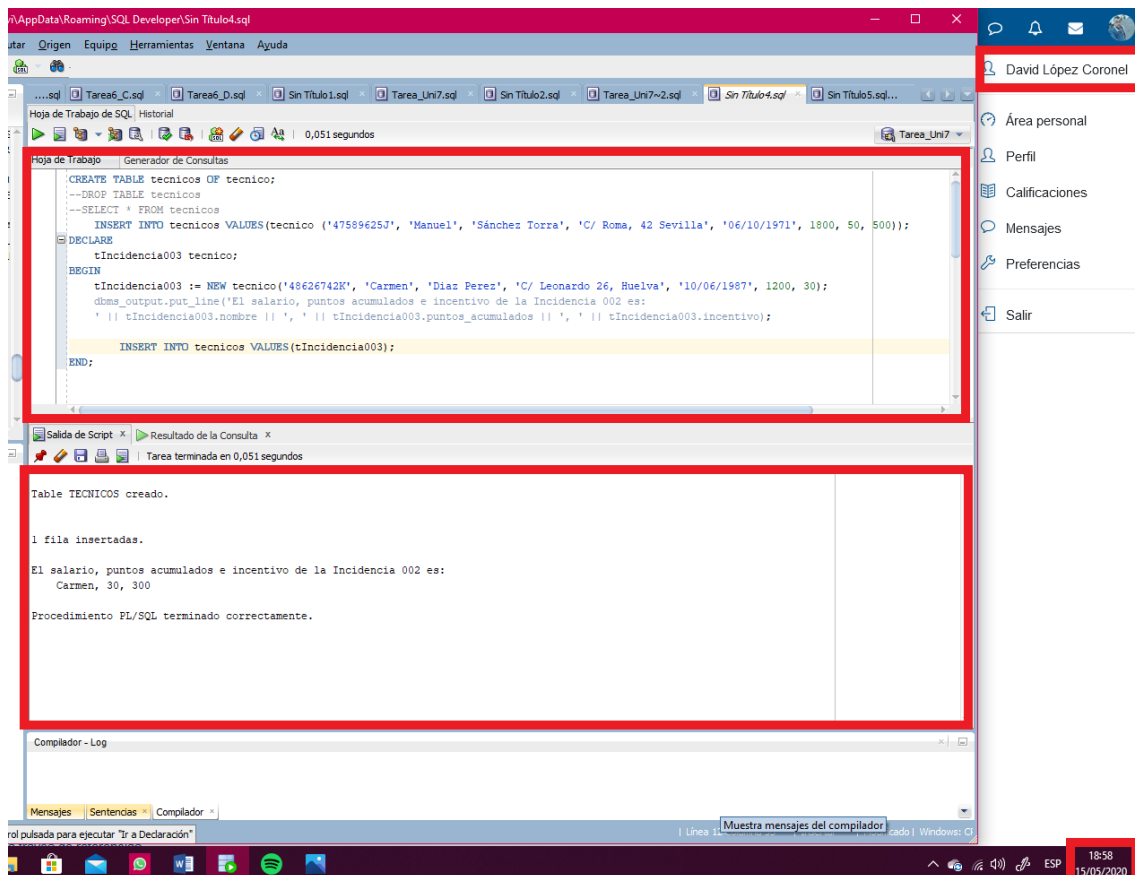
```
    dbms_output.put_line('El salario, puntos acumulados e incentivo de la  
Incidencia 002 es:
```

```
    ' || tIncidencia003.nombre || ', ' || tIncidencia003.puntos_acumulados || ', ' ||  
tIncidencia003.incentivo);
```

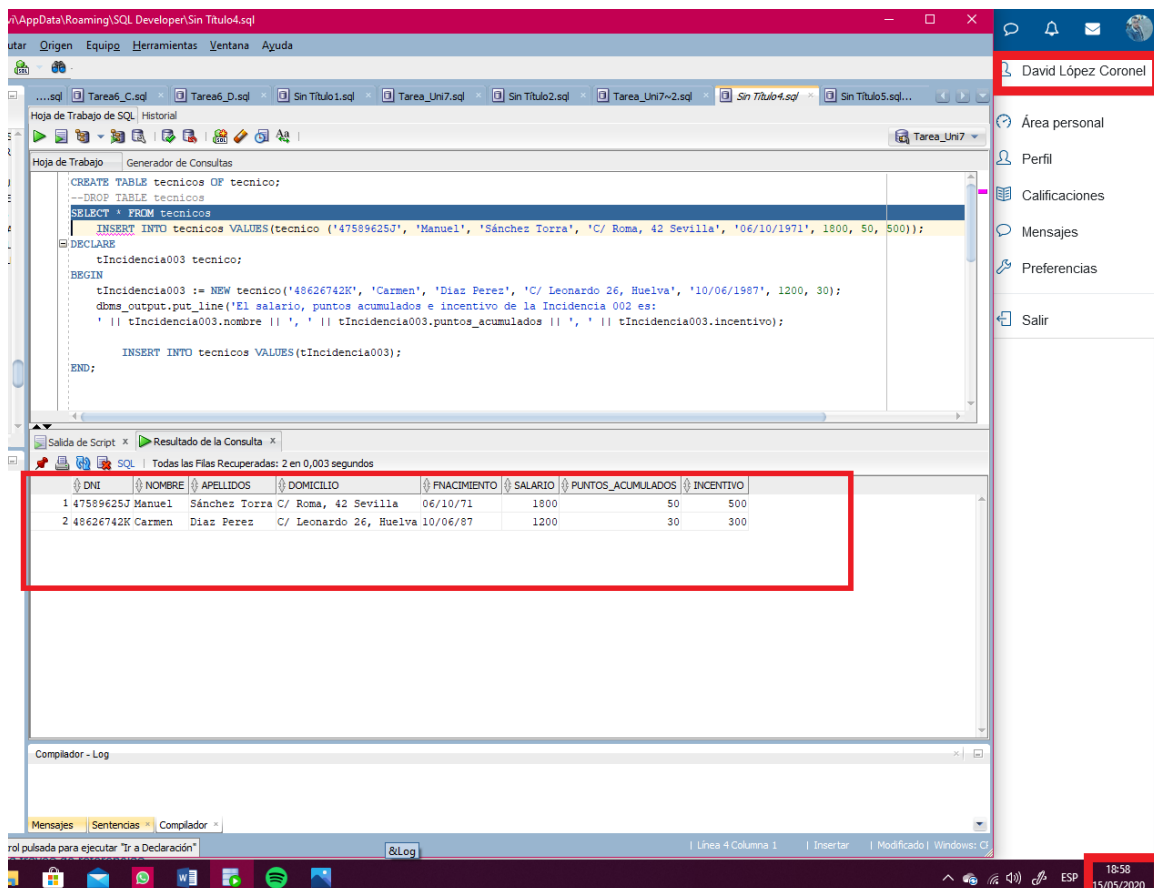
```
    INSERT INTO tecnicos VALUES(tIncidencia003);
```

```
END;
```

Capturas de pantalla correspondientes a la actividad 4;



Y en esta comprobamos que ha sido insertado el objeto o fila con los datos correspondientes al apartado a) y b).



5. Crea una colección VARRAY llamada “**ListalIncidencias**” en la que se puedan almacenar hasta 20 objetos de tipo “Incidencia”. Guarda en una instancia “**listalIncidencias1**” de dicha lista, las dos incidencias siguientes:

```
CREATE OR REPLACE TYPE ListalIncidencias IS VARRAY(20) OF incidencia;
```

```
DECLARE
```

```
    incidencia5a incidencia;
```

```
    incidencia5b incidencia;
```

```
    vreftecnico1 REF Tecnico;
```

```
    vreftecnico2 REF Tecnico;
```

```
    listalIncidencias1 ListalIncidencias;
```

```
BEGIN
```

```
    incidencia5a := NEW incidencia('501', 'Pintar piso', 'Finalizada', 3,  
    '08/10/2019', '10/10/2019', vreftecnico1);
```

```
    incidencia5b := NEW incidencia('602', 'Arreglar ducha', 'Finalizada', 20,  
    '02/05/2019', '08/05/2019', vreftecnico2);
```

```
    SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J'  
    ;
```

```
    SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI=  
    '48626742K' ;
```

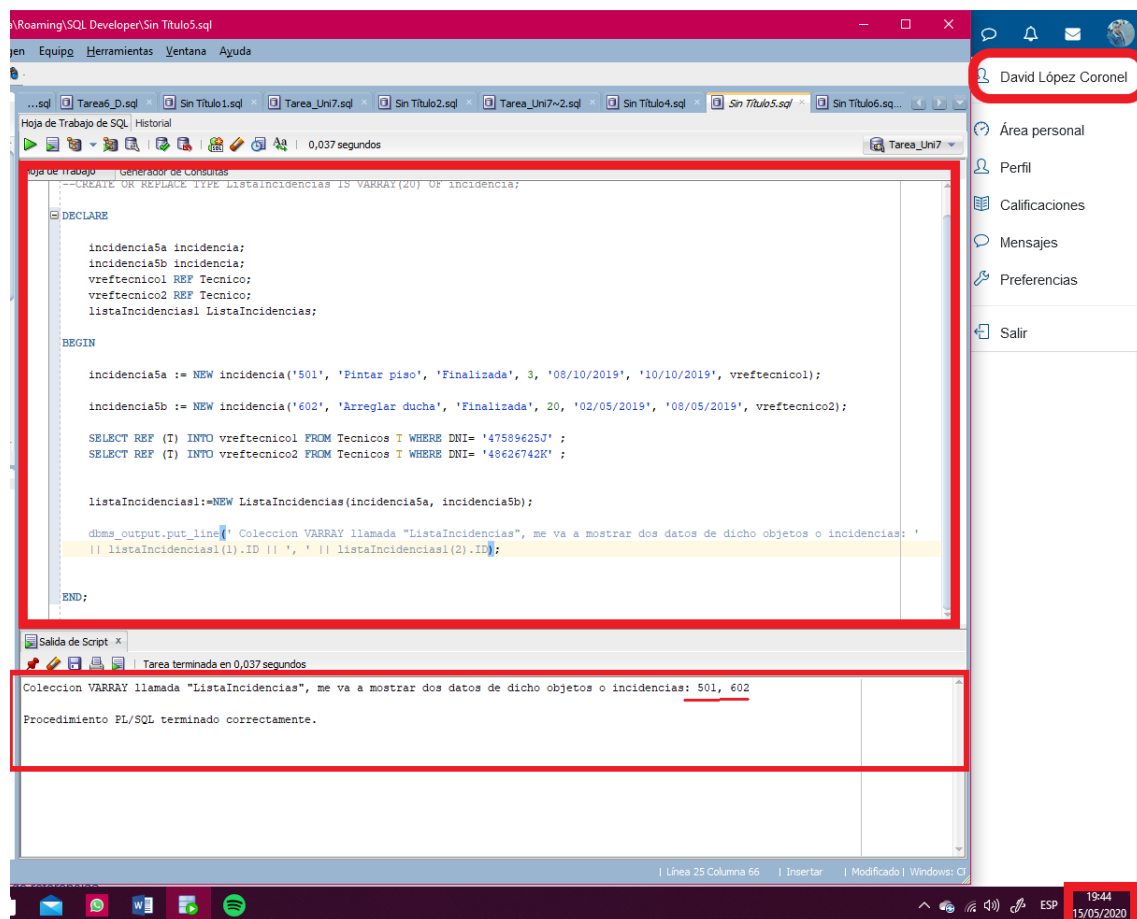
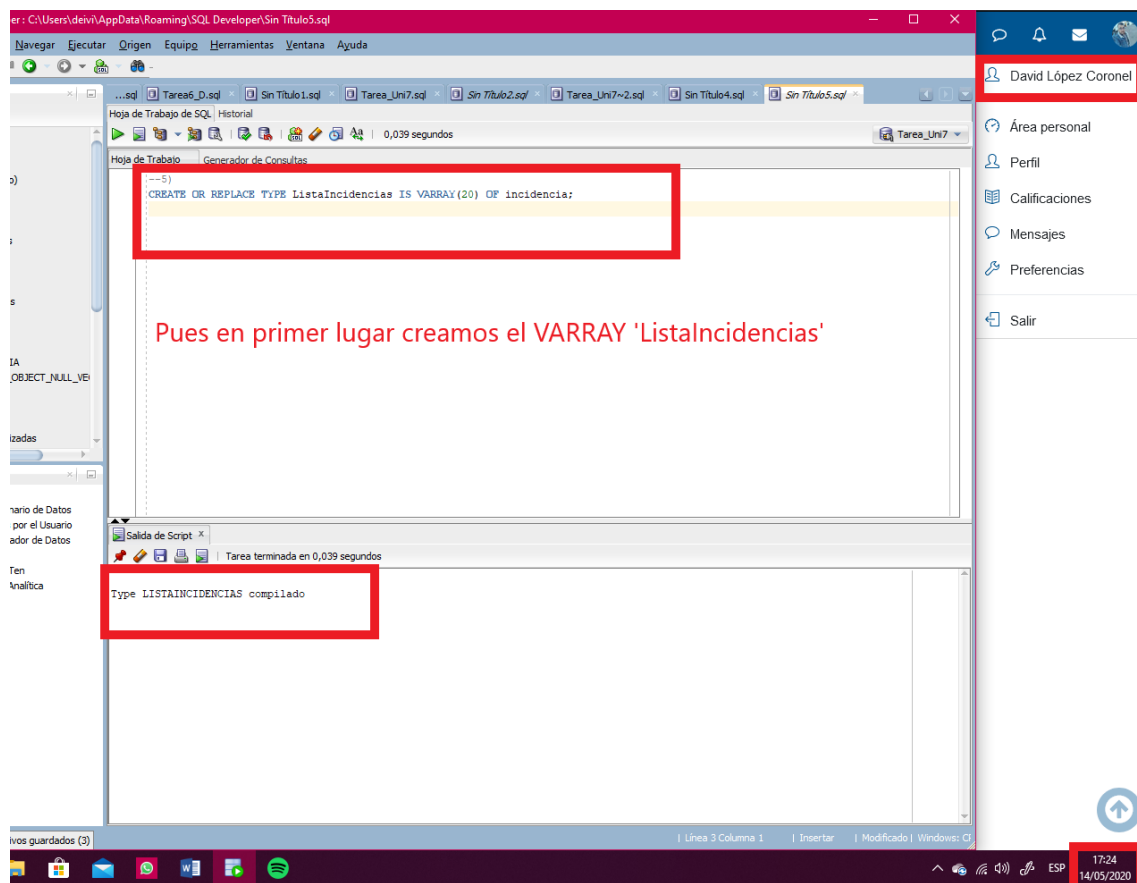
```
    listalIncidencias1:=NEW ListalIncidencias(incidencia5a, incidencia5b);
```

```
    dbms_output.put_line(' Coleccion VARRAY llamada "ListalIncidencias", me va  
    a mostrar dos datos de dicho objetos o incidencias: '
```

```
    // listalIncidencias1(1).ID // ', ' // listalIncidencias1(2).DNI);
```

```
END;
```

Capturas de pantalla de la actividad 5.



6. Crea una tabla “**Cientes**” de objetos “**Cliente**”. Inserta en dicha tabla las siguientes filas:

CREATE TABLE clientes OF cliente;

DECLARE

--Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta llamado en el objeto Cliente "incidenciaU"

vreftecnico1 REF Tecnico;

vreftecnico2 REF Tecnico;

--Instancia de cliente que llamara al objeto incidencia

IncidenciaU1 Incidencia;

IncidenciaU2 Incidencia;

BEGIN

IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3, '08/10/2019', '10/10/2019', vreftecnico1);

IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20, '02/05/2019', '08/05/2019', vreftecnico2);

SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J' ;

SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI= '48626742K' ;

INSERT INTO clientes VALUES(cliente ('25487569H', 'Pedro', 'Herrera Toledo', 'C/ Isabel II, 27 Jaén', '04/03/1979', IncidenciaU1));

INSERT INTO clientes VALUES(cliente ('35856475P', 'Vicente', 'Viedma Hidalgo', 'C/ Sorrento 22, Cádiz', '09/10/1975', IncidenciaU2));

END;

Captura correspondiente a la actividad 6.

The screenshot shows the SQL Developer interface with a PL/SQL script in the 'Hoja de Trabajo' (Worksheet) tab. The script includes a table creation, variable declarations, and an insert statement. The 'Resultado de la Consulta' (Query Result) tab at the bottom shows the message 'Procedimiento PL/SQL terminado correctamente.' (PL/SQL procedure terminated successfully).

```
--CREATE TABLE clientes OF cliente;  
SELECT * FROM clientes;  
  
DECLARE  
  --Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta llamado en el objeto Cliente "incidenciaU"  
  vreftecnico1 REF Tecnico;  
  vreftecnico2 REF Tecnico;  
  --Instancia de cliente que llamara al objeto incidencia  
  IncidenciaU1 Incidencia;  
  IncidenciaU2 Incidencia;  
  
BEGIN  
  IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3, '08/10/2019', '10/10/2019', vreftecnico1);  
  IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20, '02/05/2019', '08/05/2019', vreftecnico2);  
  
  SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '475896253' ;  
  SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI= '48626742K' ;  
  
  INSERT INTO clientes VALUES(cliente ('25487569H', 'Pedro', 'Herrera Toledo', 'C/ Isabel II, 27 Jaén', '04/03/1979', IncidenciaU1));  
  INSERT INTO clientes VALUES(cliente ('35856475P', 'Vicente', 'Viedma Hidalgo', 'C/ Sorrento 22, Cádiz', '09/10/1975', IncidenciaU2));  
  
END;
```

Procedimiento PL/SQL terminado correctamente.

En esta captura comprobamos con un SELECT, si se inserto correctamente las filas.

The screenshot shows the same SQL Developer interface, but now the 'Resultado de la Consulta' (Query Result) tab displays the data inserted into the 'clientes' table. The result is a table with 2 rows and 6 columns: DNI, NOMBRE, APELLIDOS, DOMICILIO, FINACIMIENTO, and INCIDENCIAU.

DNI	NOMBRE	APELLIDOS	DOMICILIO	FINACIMIENTO	INCIDENCIAU
1 25487569H	Pedro	Herrera Toledo	C/ Isabel II, 27 Jaén	04/03/79	[SYSTEM.INCIDENCIA]
2 35856475P	Vicente	Viedma Hidalgo	C/ Sorrento 22, Cádiz	09/10/75	[SYSTEM.INCIDENCIA]

7. Obtener de la tabla “**Cientes**”, el cliente que tiene el DNI 25487569H asignándoselo a una variable “**unCliente**”. Elimina el registro de la tabla y después utilizando esa variable modifica su incidencia asignándole la segunda incidencia que se había creado anteriormente. Inserta este cliente en la tabla “**Cientes**”.

```
CREATE TABLE clientes OF cliente;
```

```
DECLARE
```

```
--Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta  
llamado en el objeto Cliente "incidenciaU"
```

```
    vreftecnico1 REF Tecnico;
```

```
    vreftecnico2 REF Tecnico;
```

```
--Instancia de cliente que llamara al objeto incidencia
```

```
    IncidenciaU1 Incidencia;
```

```
    IncidenciaU2 Incidencia;
```

```
--Variable Cliente
```

```
    vunCliente cliente;
```

```
BEGIN
```

```
    IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3,  
'08/10/2019', '10/10/2019', vreftecnico1);
```

```
    IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20,  
'02/05/2019', '08/05/2019', vreftecnico2);
```

```
    SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J'  
    ;
```

```
    SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI=  
'48626742K' ;
```

```
    SELECT VALUE(c) INTO vunCLiente FROM Clientes c WHERE  
DNI='25487569H';
```

```
    dbms_output.put_line('Un cliente esta asignado al DNI: ' || vunCliente.DNI);
```

```
    UPDATE clientes SET vunCliente=IncidenciaU2.ID WHERE ID='510'
```

```
END;
```

Estas son las capturas correspondientes al ejercicio 7:

AppData\Roaming\SQL Developer\Sin Título7.sql

Origen Equipo Herramientas Ventana Ayuda

...sql Sin Título1.sql Tarea_Uni7.sql Sin Título2.sql Tarea_Uni7~2.sql Sin Título4.sql Sin Título5.sql Sin Título6.sql Sin Título7.sql

Hoja de Trabajo de SQL Historial

0,045 segundos

Hoja de Trabajo Generador de Consultas

```
--CREATE TABLE clientes OF cliente;
--SELECT * FROM clientes;

DECLARE
--Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta llamado en el objeto Cliente "incidenciaU"
vreftecnico1 REF Tecnico;
vreftecnico2 REF Tecnico;
--Instancia de cliente que llamara al objeto incidencia
IncidenciaU1 Incidencia;
IncidenciaU2 Incidencia;
--Variable Cliente
vunCliente cliente;

BEGIN

IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3, '08/10/2019', '10/10/2019', vreftecnico1);
IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20, '02/05/2019', '08/05/2019', vreftecnico2);

SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J' ;
SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI= '48626742K' ;

SELECT VALUE(c) INTO vunCliente FROM Clientes c WHERE DNI='25487569H';
dbms_output.put_line('Un cliente esta asignado al DNI: ' || vunCliente.DNI);

END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0,045 segundos

Procedimiento PL/SQL terminado correctamente.
Un cliente esta asignado al DNI: 25487569H

Procedimiento PL/SQL terminado correctamente.

Línea 24 Columna 81 Insertar Modificado Windows: C

AppData\Roaming\SQL Developer\Sin Título7.sql

Origen Equipo Herramientas Ventana Ayuda

...sql Sin Título1.sql Tarea_Uni7.sql Sin Título2.sql Tarea_Uni7~2.sql Sin Título4.sql Sin Título5.sql Sin Título6.sql Sin Título7.sql

Hoja de Trabajo de SQL Historial

0,026 segundos

Hoja de Trabajo Generador de Consultas

```
--CREATE TABLE clientes OF cliente;
--SELECT * FROM clientes;

DECLARE
--Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta llamado en el objeto Cliente "incidenciaU"
vreftecnico1 REF Tecnico;
vreftecnico2 REF Tecnico;
--Instancia de cliente que llamara al objeto incidencia
IncidenciaU1 Incidencia;
IncidenciaU2 Incidencia;
--Variable Cliente
vunCliente cliente;

BEGIN

IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3, '08/10/2019', '10/10/2019', vreftecnico1);
IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20, '02/05/2019', '08/05/2019', vreftecnico2);

SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J' ;
SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI= '48626742K' ;

SELECT VALUE(c) INTO vunCliente FROM Clientes c WHERE DNI='25487569H';
dbms_output.put_line('Un cliente esta asignado al DNI: ' || vunCliente.DNI);

DELETE FROM Clientes c WHERE DNI='25487569H';

END;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,026 segundos

Procedimiento PL/SQL terminado correctamente.
Un cliente esta asignado al DNI: 25487569H

Procedimiento PL/SQL terminado correctamente.

1 fila eliminado

0 filas eliminado

Línea 27 Columna 1 Insertar Modificado Windows: C

AppData\Roaming\SQL Developer\Sin Titulo7.sql

Origen Equipos Herramientas Ventana Ayuda

...sql Sin Titulo1.sql Tarea_Uni7.sql Sin Titulo2.sql Tarea_Uni7+2.sql Sin Titulo4.sql Sin Titulo5.sql Sin Titulo6.sql Sin Titulo7.sql

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```

--CREATE TABLE clientes OF cliente;
SELECT * FROM clientes;

DECLARE
--Corresponde a las REF de tecnico que tiene el objeto incidencia, el cual esta llamado en el objeto Cliente "incidenciaU"
vreftecnico1 REF Tecnico;
vreftecnico2 REF Tecnico;
--Instancia de cliente que llamara al objeto incidencia
IncidenciaU1 Incidencia;
IncidenciaU2 Incidencia;
--Variable Cliente
vunCliente cliente;

BEGIN

IncidenciaU1 := NEW Incidencia('501', 'Pintar piso', 'Finalizada', 3, '08/10/2019', '10/10/2019', vreftecnico1);
IncidenciaU2 := NEW Incidencia('602', 'Arreglar ducha', 'Finalizada', 20, '02/05/2019', '08/05/2019', vreftecnico2);

SELECT REF (T) INTO vreftecnico1 FROM Tecnicos T WHERE DNI= '47589625J' ;
SELECT REF (T) INTO vreftecnico2 FROM Tecnicos T WHERE DNI= '48626742K' ;

SELECT VALUE(c) INTO vunCliente FROM Clientes c WHERE DNI='25487569H';
dbms_output.put_line('Un cliente esta asignado al DNI: ' || vunCliente.DNI);

DELETE FROM Clientes c WHERE DNI='25487569H';

END;

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,001 segundos

DNI	NOMBRE	APELLIDOS	DOMICILIO	FNACIMIENTO	INCIDENCIAU
1 35856475P	Vicente	Viedma Hidalgo C/ Sorrento 22,	Cádiz	09/10/75	[SYSTEM.INCIDENCIA]

Linea 3 Columna 1 Insertar Modificado Windows: C

David López Coronel

Área personal

Perfil

Calificaciones

Mensajes

Preferencias

Salir

19:33 16/05/2020

David López Coronel