

# HITO 1 DEL 3º TRIMESTRE DE BASE DE DATOS

Alejandro Cortés Díaz

## Índice

INTRODUCCION .....	1
DESARROLLO .....	1
CONCLUSION.....	13
<b>Enlace a GitHub</b> .....	14
<b>Bibliografía</b> .....	15

## INTRODUCCION

- En esta actividad he realizado una conexión a una bdd para simular un trabajo donde debía crear un tipo objeto tSocio, con atributos y métodos propios, así como una primary key en la tabla socios relacionado con el tipo objeto dicho. Y lo mismo con un tipo objeto tLibro, y su tabla, que contaba con foreign key de la tabla Socio (NIF). Realicé inserciones en ambas tablas, y ejecuté selects donde empleaba las funciones descritas en el archivo SQL, para mostrar datos de diferentes formas.

## DESARROLLO

- Crea el tipo objeto tSocio con los atributos que muestra la imagen y un método llamado getSocio() que devolverá una cadena con la concatenación de todos los datos personales del socio. 1 punto.
- Esta es una función donde devuelvo un tipo varchar con los valores de las variables previamente definidas en el tipo objeto tSocio, concatenando los valores con apartados String que presentan lo que se va a mostrar, y a continuación, con separadores, extraigo el valor de la variable asignada en el insert into

```
--1

CREATE OR REPLACE TYPE tSocio AS OBJECT (
    NIF VARCHAR2(20),
    Nombre VARCHAR2(100),
    Tlf VARCHAR2(20),
    MEMBER FUNCTION getSocio RETURN VARCHAR2
);
/
CREATE OR REPLACE TYPE BODY tSocio AS
MEMBER FUNCTION getSocio RETURN VARCHAR2 IS
BEGIN
    RETURN 'NIF: ' || NIF || ', Nombre: ' || Nombre || ', Teléfono: ' || Tlf;
END;
END;
/
```

Salida de Script x

Tarea terminada en 0,897 segundos

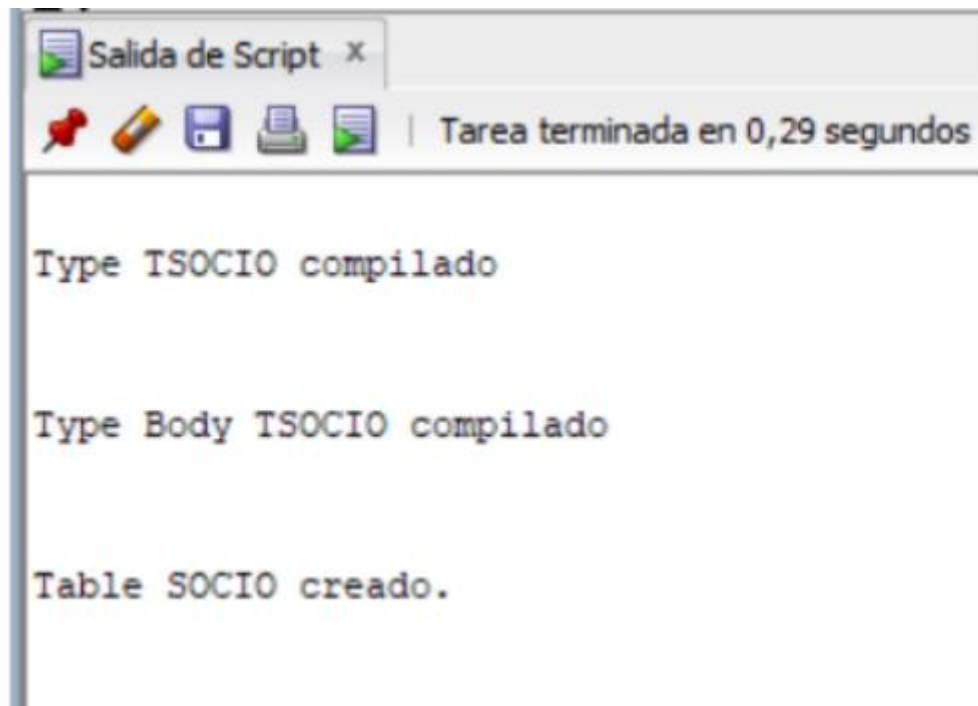
Type ISOCIO compilado

Type Body ISOCIO compilado

- 2. Crea una tabla llamada Socio de objetos de tipo tSocio. No olvides definir el atributo NIF como Primary Key. 1 punto.
- Aquí simplemente creo la tabla a partir del tipo objeto tSocio con la primary key

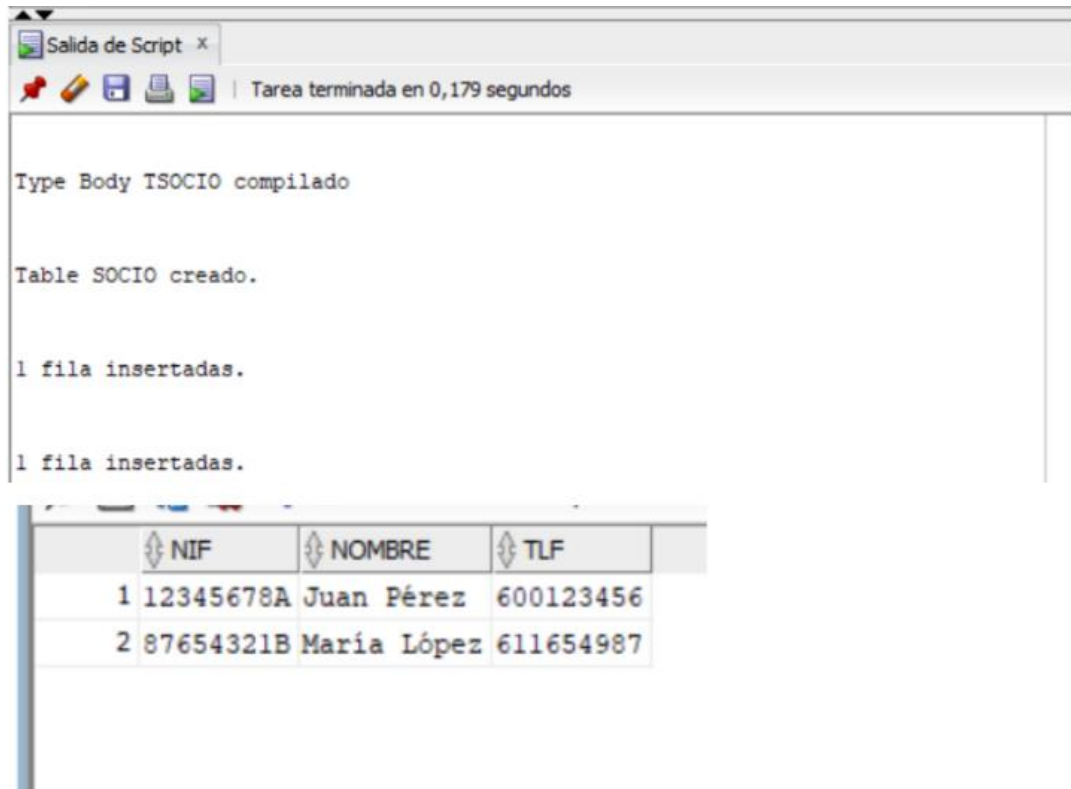
```
-- 2

CREATE TABLE Socio OF tSocio (
    NIF PRIMARY KEY
);
```



- 3. Añade 2 socios a la tabla Socios. 1 punto.
- Realizo las inserciones cuidando respetar los tipos de datos asignados a cada uno

```
-- 3  
  
INSERT INTO Socio VALUES (tSocio('12345678A', 'Juan Pérez', '600123456'));  
INSERT INTO Socio VALUES (tSocio('87654321B', 'María López', '611654987'));  
  
-- 4
```



- 4. Crea un tipo objeto llamado tLibro con los atributos que muestra la imagen y los siguientes métodos:
- getLibro() que retorna una cadena con todos los datos del libro.
- diasPrestamo() que retorna un número que indica los días que lleva el libro prestado. Se calcula con la fecha de préstamo y la fecha actual.
- diasMulta() calculados suponiendo que el libro se presta solo por 7 días. 3 puntos.

- Aquí tuve algunos problemas con las funciones ya que el resto era simplemente encontrar el tipo de dato adecuado, y observar si era la sintaxis adecuada a utilizar.
- Sobre todo con diasPrestamo, ya que no recordaba como tener en cuenta la fecha que proporcionaba el sistema y pensé que era algo mucho más complejo que usar SysDate, que ya de por sí lo alberga.
- Con diasMulta pasó algo parecido, puesto que me costaba sacar la lógica por la cual debía hacer la resta porque la multa no se aplicaría sino. Aunque el uso de TRUNC para filtrar los datos del SysDate, y rechazar las horas, fue la parte más complicada. Puesto que GREATEST simplemente lo usé para que no diera valor negativo.

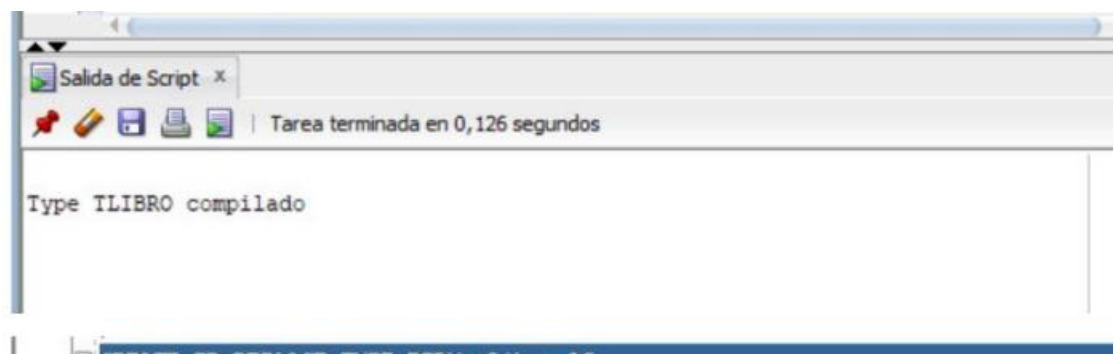
```

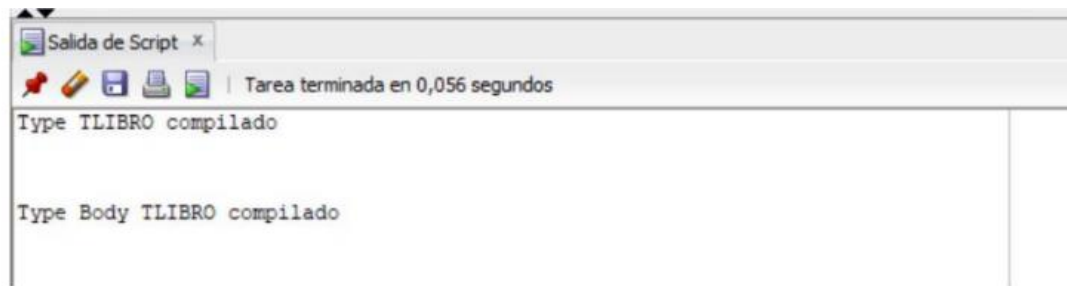
CREATE OR REPLACE TYPE tLibro AS OBJECT (
    Referencia NUMBER,
    Titulo VARCHAR2(100),
    Autor VARCHAR2(100),
    Editorial VARCHAR2(100),
    FechaPrestamo DATE,
    NIFSocio VARCHAR2(20),
    MEMBER FUNCTION getLibro RETURN VARCHAR2,
    MEMBER FUNCTION diasPrestamo RETURN NUMBER,
    MEMBER FUNCTION diasMulta RETURN NUMBER
);
/
CREATE OR REPLACE TYPE BODY tLibro AS
MEMBER FUNCTION getLibro RETURN VARCHAR2 IS
BEGIN
    RETURN 'Referencia: ' || Referencia || ', Título: ' || Titulo ||
        ', Autor: ' || Autor || ', Editorial: ' || Editorial;
END;

MEMBER FUNCTION diasPrestamo RETURN NUMBER IS
BEGIN
    RETURN TRUNC(SYSDATE - FechaPrestamo);
END;

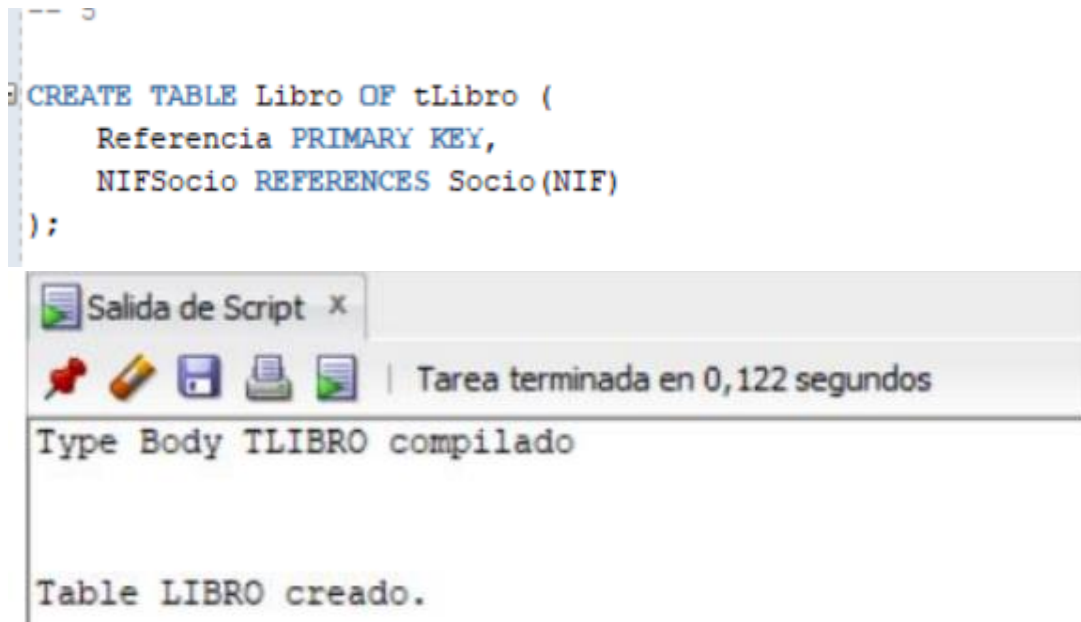
MEMBER FUNCTION diasMulta RETURN NUMBER IS
BEGIN
    RETURN GREATEST(0, TRUNC(SYSDATE - FechaPrestamo) - 7);
END;
END;
/

```





- 5. Crea una tabla llamada Libro de objetos de tipo tLibro. No olvides definir el atributo Referencia como Primary key y NIFSocio como Foreign Key que permite asociar cada libro con el socio que lo tiene alquilado. 1 punto.
- Creo la tabla Libro del tipo objeto tLibro y establezco de foreign key la primary de la tabla Socio, a su vez, del tipo objeto tSocio.



- 6. Añade 4 libros. 1 punto.
- Aquí la dificultad que pude encontrar fue el uso del TO\_DATE y la necesidad de aplicar el formato de fecha inverso al que pensé que era por costumbre.

-- 6

```
INSERT INTO Libro VALUES (tLibro(1, 'Cien años de soledad', 'Gabriel Garcia Márquez', 'Sudamericana', TO_DATE('2024-03-25', 'YYYY-MM-DD'), '12345678A'));
INSERT INTO Libro VALUES (tLibro(2, '1984', 'George Orwell', 'Secker y Warburg', TO_DATE('2024-03-20', 'YYYY-MM-DD'), '87654321B'));
INSERT INTO Libro VALUES (tLibro(3, 'El principito', 'Antoine de Saint-Exupéry', 'Reynal y Hitchcock', TO_DATE('2024-03-30', 'YYYY-MM-DD'), '87654321B'));
INSERT INTO Libro VALUES (tLibro(4, 'Don Quijote', 'Miguel de Cervantes', 'Francisco de Robles', TO_DATE('2024-03-18', 'YYYY-MM-DD'), '12345678A'));
```

REFERENCIA	TITULO	AUTOR	EDITORIAL	FECHAPRESTAMO	NIFSOCIO
1	1 Cien años de soledad	Gabriel Garcia Márquez	Sudamericana	25/03/24	12345678A
2	2 1984	George Orwell	Secker y Warburg	20/03/24	87654321B
3	3 El principito	Antoine de Saint-Exupéry	Reynal y Hitchcock	30/03/24	87654321B
4	4 Don Quijote	Miguel de Cervantes	Francisco de Robles	18/03/24	12345678A

- 7. Ejecuta 8 SELECTs que te parezcan de utilidad para recuperar datos de la base de datos. Indica lo que pretendes obtener en cada una de ellas. 2 puntos.

```
-- 7

-- La cantidad de socios que tienen libros, y cuántos
SELECT S.NIF, S.Nombre, COUNT(L.Referencia) AS cantidad_libros
FROM Socio S
LEFT JOIN Libro L ON S.NIF = L.NIFSocio
GROUP BY S.NIF, S.Nombre;

-- Muestra todos los libros que han sido prestados
SELECT L.getLibro() AS DetallesLibro
FROM Libro L;

-- Cuántos días llevan prestados los libros y cuántos días de multa llevan acumulados estos
SELECT L.Titulo, L.diasPrestamo() AS Dias_Prestamo, L.diasMulta() AS Dias_Multa
FROM Libro L;

-- Uso 2 métodos para mostrar los detalles del socio con el libro solicitado
SELECT S.getSocio() AS DetallesSocio, L.getLibro() AS DetallesLibro
FROM Socio S
INNER JOIN Libro L ON S.NIF = L.NIFSocio;

-- Todos los libros que tienen al menos 1 día de multa (todos tienen por los datos del insert into)
SELECT L.Titulo, L.diasPrestamo() AS Dias_Prestamo, L.diasMulta() AS Dias_Multa
FROM Libro L
WHERE L.diasMulta() > 0;

-- Muestra los libros, su autor, y la fecha en la que fueron prestados
SELECT L.Titulo, L.Autor, L.FechaPrestamo
FROM Libro L
ORDER BY L.FechaPrestamo ASC;

-- Obtiene todos los días en total que han estado prestados los libros en su conjunto
SELECT SUM(L.diasPrestamo()) AS Total_Dias_Prestamo
FROM Libro L;

-- Muestra los libros que lleven días de multa, junto con el nombre y NIF del cliente que posee dichos libros
SELECT S.NIF, S.Nombre, L.Titulo, L.diasMulta() AS DiasMulta
FROM Socio S
JOIN Libro L ON S.NIF = L.NIFSocio
WHERE L.diasMulta() > 0;
```





- Aquí el left join fue lo más significativo, es una función que puede confundirme en algunos casos y tuve que mirar ejemplos de W3School

-- La cantidad de socios que tienen libros, y cuántos

```
SELECT S.NIF, S.Nombre, COUNT(L.Referencia) AS cantidad_libros
```

```
FROM Socio S
```

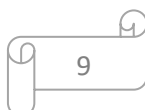
```
LEFT JOIN Libro L ON S.NIF = L.NIFSocio
```

```
GROUP BY S.NIF, S.Nombre;
```



Microsoft Word

	NIF	NOMBRE	CANTIDAD_LIBROS
1	12345678A	Juan Pérez	2
2	87654321B	María López	2



2-

- Esta no me resultó compleja porque solamente tienes que llamar al método y especificar de donde viene

-- Muestra todos los libros que han sido prestados

```
SELECT L.getLibro() AS DetallesLibro
```

```
FROM Libro L;
```

DETALLESLIBRO
1 Referencia: 1, Título: Cien años de soledad, Autor: Gabriel Garcia Márquez, Editorial: Sudamericana
2 Referencia: 2, Título: 1984, Autor: George Orwell, Editorial: Secker y Warburg
3 Referencia: 3, Título: El principito, Autor: Antoine de Saint-Exupéry, Editorial: Reynal y Hitchcock
4 Referencia: 4, Título: Don Quijote, Autor: Miguel de Cervantes, Editorial: Francisco de Robles

3-

- Aplica algo parecido con esto, aunque en este caso son 2 llamadas de dos métodos distintos con 2 alias

-- Cuántos días llevan prestados los libros y cuántos días de multa llevan acumulados estos

```
SELECT L.Titulo, L.diasPrestamo() AS Dias_Prestamo, L.diasMultas() AS Dias_Multa
```

```
FROM Libro L;
```

TITULO	DIAS_PRESTAMO	DIAS_MULTA
1 Cien años de soledad	375	368
2 1984	380	373
3 El principito	370	363
4 Don Quijote	382	375

4-

- Aquí y en esta y algunas siguientes juego con el conjunto de datos mostrados, ya que en algunos contextos será más o menos interesante tener unos u otros datos acompañados, en distintas combinaciones de contenidos

-- Uso 2 métodos para mostrar los detalles del socio con el libro solicitado

```
SELECT S.getSocio() AS DetallesSocio, L.getLibro() AS DetallesLibro
```

```
FROM Socio S
```

```
INNER JOIN Libro L ON S.NIF = L.NIFSocio;
```

DETALLESSOCIO	DETALLESLIBRO
1 NIF: 12345678A, Nombre: Juan Pérez, Teléfono: 600123456	Referencia: 1, Título: Cien años de soledad, Autor: Gabriel Garcia Márquez, Editorial: Sudamericana
2 NIF: 87654321B, Nombre: María López, Teléfono: 611654987	Referencia: 2, Título: 1984, Autor: George Orwell, Editorial: Secker y Warburg
3 NIF: 87654321B, Nombre: María López, Teléfono: 611654987	Referencia: 3, Título: El principito, Autor: Antoine de Saint-Exupéry, Editorial: Reynal y Hitchcock
4 NIF: 12345678A, Nombre: Juan Pérez, Teléfono: 600123456	Referencia: 4, Título: Don Quijote, Autor: Miguel de Cervantes, Editorial: Francisco de Robles

5-

- Este select me parece importante aunque en mi caso todos tenían al menos 1 día de multa, pero es un filtrado esencial aunque sencillo considero.

-- Todos los libros que tienen al menos 1 día de multa (todos tienen por los datos del insert into)

```
SELECT L.Titulo, L.diasPrestamo() AS Dias_Prestamo, L.diasMulta() AS Dias_Multa
```

```
FROM Libro L
```

```
WHERE L.diasMulta() > 0;
```

	TITULO	DIAS_PRESTAMO	DIAS_MULTA
1	Cien años de soledad	375	368
2	1984	380	373
3	El principito	370	363
4	Don Quijote	382	375

6-

- Uno de los ejemplos de entrega de datos con información complementaria útil, puedes saber de esta forma cuales son los libros que tienen más días acumulados con respecto a aquellos que han estado prestados.

-- Muestra los libros, su autor, y la fecha en la que fueron prestados

```
SELECT L.Titulo, L.Autor, L.FechaPrestamo
```

```
FROM Libro L
```

```
ORDER BY L.FechaPrestamo ASC;
```

	TITULO	DIAS_PRESTAMO	DIAS_MULTA
1	Cien años de soledad	375	368
2	1984	380	373
3	El principito	370	363
4	Don Quijote	382	375

7-

- Esto nos permitiría conocer cuál es el valor de actividad con nuestros libros, podría ser especialmente útil para comparar rendimiento de entregas o prestamos a lo largo de los meses/años

-- Obtiene todos los días en total que han estado prestados los libros en su conjunto

```
SELECT SUM(L.diasPrestamo()) AS Total_Dias_Prestamo
```

```
FROM Libro L;
```

	TOTAL_DIAS_PRESTAMO
1	1507

8-

- Esta está más orientada a tener una información más completa orientada al cliente con su nif, este select puede ser interesante si queremos saber directamente la identificación de las personas con multa, para poder comunicar con ellas inmediatamente después, por ejemplo, o enviar un aviso con el NIF de referencia, para que salga algún error cuando trate de obtener otro libro, por precaución.

-- Muestra los libros que lleven dias de multa, junto con el nombre y NIF del cliente que posee dichos libros

```
SELECT S.NIF, S.Nombre, L.Titulo, L.diasMulta() AS DiasMulta
```

```
FROM Socio S
```

```
JOIN Libro L ON S.NIF = L.NIFSocio
```

```
WHERE L.diasMulta() > 0;
```

⚡ NIF	⚡ NOMBRE	⚡ TITULO	⚡ DIASMULTA
1 12345678A	Juan Pérez	Cien años de soledad	368
2 87654321B	María López 1984		373
3 87654321B	María López	El principito	363
4 12345678A	Juan Pérez	Don Quijote	375

## CONCLUSION

- Ha sido una actividad interesante y sencilla, prefiero que sean así porque a la par de hacerte sentir más realizado, supone una altura realista a la que llegar, teniendo en cuenta que contamos con un tiempo bastante más reducido para su desarrollo

## Enlace a GitHub

<https://github.com/Cortes-cmd/BBDD.git>

## Bibliografía

Alcalde, A. (2016, enero 1). *PL/SQL. Procedimientos y Funciones*.  
PL/SQL. Procedimientos y Funciones.  
<http://elbaultdelprogramador.com/plsql-procedimientos-y-funciones/>

Bellido Romero, A. M. (2019, enero 21). *Crear una tabla de objetos en Oracle*. Blogspot.com. <http://ambellido.blogspot.com/2019/01/crear-una-tabla-de-objetos-en-oracle.html>

*GREATEST*. (s/f). Recuperado el 10 de abril de 2025, de  
<https://docs.oracle.com/en/database/oracle/database/21/sqlrf/GREATEST.html>

*Obtener en una variable el resultado de un select en oracle*. (s/f).  
Forosdelweb.com. Recuperado el 10 de abril de 2025, de  
<https://www.forosdelweb.com/f100/obtener-variable-resultado-select-oracle-1088058/>

*Sintaxis para Hacer Referencia a Variables*. (s/f). Oracle.com. Recuperado el 10 de abril de 2025, de  
<https://docs.oracle.com/middleware/12211help/bicc/es/bi.12211/e73374/GUID-F2718160-11E3-4CA4-852A-5095F55EBE29.htm>

*Truncate/ Trunc*. (s/f). Oracle.com. Recuperado el 10 de abril de 2025, de  
[https://docs.oracle.com/cloud/help/es/freeform/FREEF/truncate\\_trunc.htm](https://docs.oracle.com/cloud/help/es/freeform/FREEF/truncate_trunc.htm)