**HITO 1 DEL 3º TRIMESTRE DE Bases de datos**



* Mario Sanz
* Fecha de entrega: 10 / 4 / 2025

Contenido

[Repositorio de GitHub 2](#_Toc194570380)

[Desarrollo de la actividad 2](#_Toc194570381)

[Creación de la base de datos y el usuario: 2](#_Toc194570382)

[Nos conectamos al usuario que hemos creado: 3](#_Toc194570383)

[Empezamos a modificar la base de datos, creación de objeto tsocio y su método: 3](#_Toc194570384)

[Creamos el objeto y la tabla de los libros: 4](#_Toc194570385)

[Bibliografía 5](#_Toc194570386)

# Repositorio de GitHub

<https://github.com/mariosanz910/HITO-1-DEL-3-TRIMESTRE-DE-Bases-de-datos-Mario-Sanz.git>

# Desarrollo de la actividad

## Creación de la base de datos y el usuario:

-- CREA UNA BASE DE DATOS PLUGGABLE (PDB) LLAMADA "BIBLIOTECA"

CREATE PLUGGABLE DATABASE biblioteca

ADMIN USER admin IDENTIFIED BY admin\_password

FILE\_NAME\_CONVERT = ('opt/oracle/oradata/FREE/pdbseed/', 'opt/oracle/oradata/FREE/pdb\_biblioteca/');

-- MUESTRA EL NOMBRE DE LA BASE DE DATOS ACTUAL

SHOW CON\_NAME;

-- CAMBIA AL CONTEXTO DE LA BASE DE DATOS PLUGGABLE "BIBLIOTECA"

ALTER SESSION SET CONTAINER = biblioteca;

-- MUESTRA LAS BASES DE DATOS PLUGGABLES DISPONIBLES

SHOW PDBS;

-- ABRE LA BASE DE DATOS PLUGGABLE "BIBLIOTECA" PARA SU USO

ALTER PLUGGABLE DATABASE biblioteca OPEN;

-- CREA UN USUARIO LLAMADO "BIBLIOTECA\_ADMIN\_MSV" CON CONTRASEÑA "Biblioteca1234"

CREATE USER biblioteca\_admin\_msv IDENTIFIED BY Biblioteca1234;

-- CONCEDE PRIVILEGIOS DE CONEXIÓN Y RECURSOS AL USUARIO

GRANT CONNECT, RESOURCE TO biblioteca\_admin\_msv;

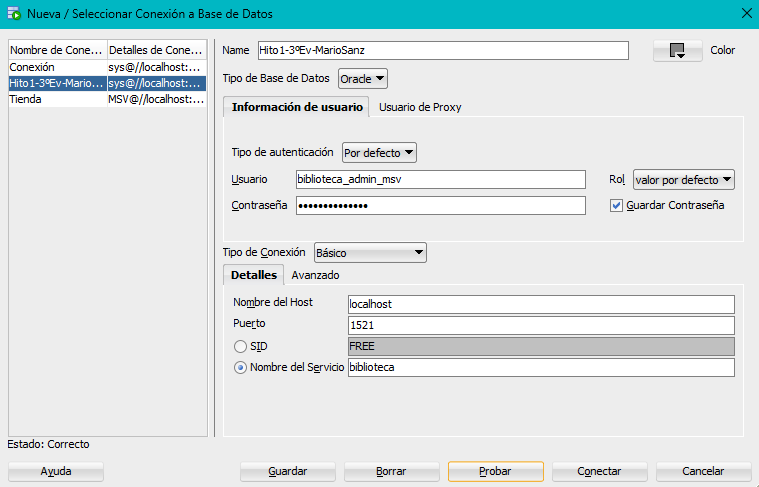
-- OTORGA PERMISO PARA USAR ESPACIO ILIMITADO EN TABLESPACES

GRANT UNLIMITED TABLESPACE TO biblioteca\_admin\_msv;

-- ASIGNA PRIVILEGIOS PARA CREAR SESIONES, TABLAS, VISTAS, PROCEDIMIENTOS, SECUENCIAS Y TRIGGERS

GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER TO biblioteca\_admin\_msv;

## Nos conectamos al usuario que hemos creado:





## Empezamos a modificar la base de datos, creación de objeto tsocio y su método:

CREATE OR REPLACE TYPE tSocio AS OBJECT (

NIF VARCHAR2(9),

Nombre VARCHAR2(100),

Telefono VARCHAR2(15),

-- MÉTODO getSocio que devolverá todos los datos de Socio

MEMBER FUNCTION getSocio RETURN VARCHAR2

);



CREATE OR REPLACE TYPE BODY tSocio AS

MEMBER FUNCTION getSocio RETURN VARCHAR2 IS

BEGIN

-- DEVUELVE UNA CADENA CON LA CONCATENACIÓN DE LOS DATOS PERSONALES DEL SOCIO

RETURN 'NIF: ' || NIF || ', Nombre: ' || Nombre || ', Teléfono: ' || Telefono;

END;

END;



CREATE TABLE Socio OF tSocio (

NIF PRIMARY KEY

);

-- CREAMOS UNA TABLA A PARTIR DE LOS DATOS DEL OBJETO TSOCIO



INSERT INTO Socio VALUES (tSocio('12345678A', 'Mario Sanz', '123456789'));

INSERT INTO Socio VALUES (tSocio('87654321B', 'Ricardo Perez', '987654321'));

## Creamos el objeto y la tabla de los libros:

DROP TABLE IF EXISTS Libro;

CREATE OR REPLACE TYPE tLibro AS OBJECT (

Referencia VARCHAR2(20),

Titulo VARCHAR2(100),

Autor VARCHAR2(100),

Editorial VARCHAR2(100),

FechaPrestamo DATE,

NIFSocio VARCHAR2(9),

MEMBER FUNCTION getLibro RETURN VARCHAR2,

MEMBER FUNCTION diasPrestamo RETURN NUMBER,

MEMBER FUNCTION diasMulta RETURN NUMBER

);



-- Crear el cuerpo del tipo tLibro

CREATE OR REPLACE TYPE BODY tLibro AS

MEMBER FUNCTION getLibro RETURN VARCHAR2 IS

BEGIN

RETURN 'Referencia: ' || Referencia || ', Título: ' || Titulo || ', Autor: ' || Autor ||

', Editorial: ' || Editorial || ', Fecha de Préstamo: ' || TO\_CHAR(FechaPrestamo, 'DD/MM/YYYY') ||

', NIF Socio: ' || NIFSocio;

END;

MEMBER FUNCTION diasPrestamo RETURN NUMBER IS dias NUMBER;

BEGIN

-- Calcular los días que lleva prestado el libro

dias := SYSDATE - FechaPrestamo;

RETURN dias;

END;

MEMBER FUNCTION diasMulta RETURN NUMBER IS dias NUMBER;

BEGIN

-- Calcular los días de multa (si pasaron más de 7 días desde el préstamo)

dias := SYSDATE - FechaPrestamo;

IF dias > 7 THEN

RETURN dias - 7; -- Días de multa

ELSE

RETURN 0; -- No hay multa si el préstamo no supera los 7 días

END IF;

END;

END;

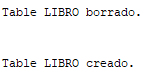


CREATE TABLE Libro OF tLibro (

Referencia PRIMARY KEY,

NIFSocio REFERENCES Socio(NIF)

);



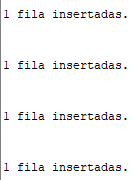
INSERT INTO Libro VALUES (tLibro('l1', 'La República', 'Platón', 'Editorial A', TO\_DATE('2025-03-01', 'YYYY-MM-DD'), '87654321B'));

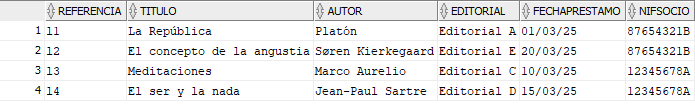
INSERT INTO Libro VALUES (tLibro('l2', 'El concepto de la angustia', 'Søren Kierkegaard', 'Editorial E', TO\_DATE('2025-03-20', 'YYYY-MM-DD'), '87654321B'));

INSERT INTO Libro VALUES (tLibro('l3', 'Meditaciones', 'Marco Aurelio', 'Editorial C', TO\_DATE('2025-03-10', 'YYYY-MM-DD'), '12345678A'));

INSERT INTO Libro VALUES (tLibro('l4', 'El ser y la nada', 'Jean-Paul Sartre', 'Editorial D', TO\_DATE('2025-03-15', 'YYYY-MM-DD'), '12345678A'));

SELECT \* FROM libro;





## Selects:

SELECT \* FROM Socio;

/\*Esta consulta devuelve todos los datos de los socios almacenados en la base de datos.

Es útil para obtener un listado completo de los socios que están registrados,

lo cual es fundamental para conocer quiénes son los usuarios del sistema.\*/



SELECT \* FROM Socio WHERE NIF = '12345678A';

/\*Permite buscar la información de un socio en particular mediante su NIF.

Esto es útil para acceder rápidamente a los datos de un socio cuando se necesita verificar su información,

como su nombre o teléfono, para realizar un seguimiento de sus préstamos.\*/



SELECT Referencia, Titulo, FechaPrestamo

FROM Libro

WHERE FechaPrestamo >= SYSDATE - 10;

/\*Esta consulta devuelve todos los libros que han sido prestados en los últimos 10 días.

Es útil para gestionar y realizar un seguimiento de los préstamos más recientes,

lo cual es relevante para saber qué libros han sido más solicitados últimamente.

Se puede cambiar el 10 por una cantidad de dias deseada\*/



SELECT NIFSocio, COUNT(\*) AS NumLibros

FROM Libro

GROUP BY NIFSocio

HAVING COUNT(\*) > 1;

/\*Esta consulta identifica a los socios que tienen más de un libro prestado.

Es útil para monitorear a los socios que son usuarios frecuentes y para garantizar

que los libros se devuelvan a tiempo, evitando el abuso del sistema de préstamos.\*/



SELECT Libro.Referencia, Libro.Titulo, Libro.Autor, Socio.Nombre, Socio.Telefono

FROM Libro

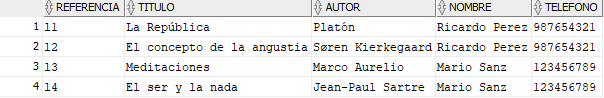
JOIN Socio ON Libro.NIFSocio = Socio.NIF;

/\*Esta consulta recupera todos los libros junto con los detalles del socio que los ha tomado prestados.

Relaciona las tablas Libro y Socio mediante la clave foránea NIFSocio en la tabla Libro.

Es útil para obtener un listado completo de libros prestados

junto con la información de contacto de los socios que los han tomado.\*/



SELECT Libro.Titulo, Socio.Nombre

FROM Libro

JOIN Socio ON Libro.NIFSocio = Socio.NIF

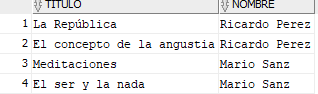
WHERE Libro.FechaPrestamo <= SYSDATE;

/\*Esta consulta muestra los títulos de los libros y los nombres de los socios que han tomado un libro prestado.

Utiliza la fecha actual (SYSDATE) para identificar los libros que ya han sido prestados

(asumiendo que los libros que se encuentran en la tabla son los que aún no han sido devueltos).

Es útil para identificar los libros actualmente prestados y saber qué socios los tienen.\*/



SELECT Referencia, Titulo, FechaPrestamo

FROM Libro

WHERE Autor = 'Platón';

/\*Este SELECT obtiene los libros prestados de Platón, mostrando su referencia,

título y fecha de préstamo. La consulta es más directa y simple, manteniendo solo

lo esencial para obtener la información requerida.\*/



SELECT Nombre, Telefono, COUNT(\*) AS NumeroDeLibros

FROM Socio s, Libro l

WHERE s.NIF = l.NIFSocio

AND l.FechaPrestamo < SYSDATE

GROUP BY Nombre, Telefono

HAVING COUNT(\*) > 2

ORDER BY NumeroDeLibros DESC;

/\*Este SELECT sigue siendo funcional pero usa una sintaxis más simple al eliminar

el JOIN explícito y usar una forma más directa de juntar las tablas Socio y Libro con la condición en el WHERE.

También se mantiene la lógica de contar los libros prestados que ya tienen una

fecha de préstamo pasada y filtrar solo aquellos socios con más de dos libros prestados.\*/

# Bibliografía

1. Oracle. (2025). Oracle Database Documentation. <https://docs.oracle.com/en/database/>
2. Date, C. J. (2012). SQL and Relational Theory: How to Write Accurate SQL Code (2nd ed.). O'Reilly Media.
3. Oracle. (2025). PL/SQL Procedural Language. <https://docs.oracle.com/cd/B28359_01/appdev.111/b31254/toc.htm>
4. Oracle. (2025). Oracle SQL Language Reference. <https://docs.oracle.com/cd/B28359_01/server.111/b28286/toc.htm>
5. Oracle. (2025). Oracle Database Concepts. <https://docs.oracle.com/cd/B28359_01/server.111/b28318/toc.htm>