



# **TUTORIAL DE ORACLE CON DBEAVER**

Laura Juncos Leal. 2ºDAM

Lo primero que vamos a hacer es instalar DBeaver del sitio oficial:

<https://dbeaver.io/download/>

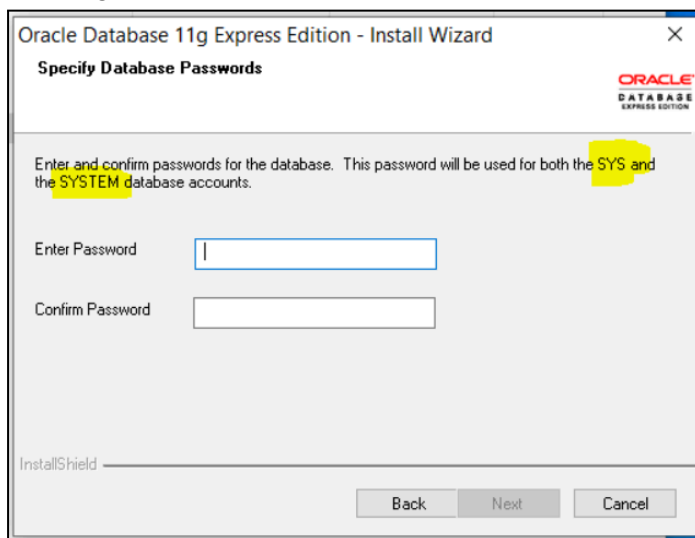
Instalaremos DBeaver Community 24.3.3.

Una vez que lo tengamos instalado nos tendremos que instalar Oracle DataBase 21c:

<https://www.oracle.com/es/database/technologies/xe-downloads.html>

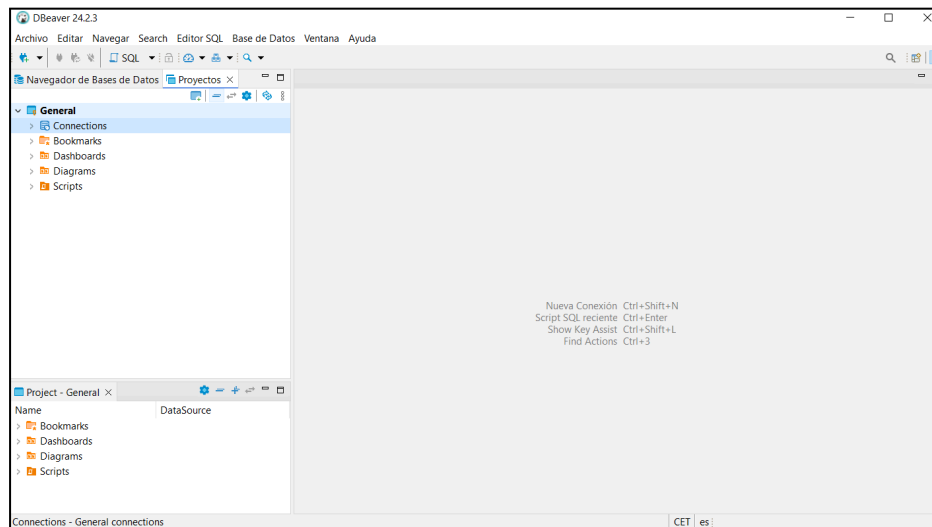
Edición Express de Oracle Database 21c	
Descargar	Descripción
 <a href="#">Oracle Database 21c Express Edition para Windows x64</a>	(1,967,615,483 bytes - 8 de octubre de 2021) [Sha256sum: 939742c3305c466566a55f607638621b6aa7033a183175f6bcd6cffb48e6bc3f]

Descargarlo e instalarlo.



Cuando ya lo tengamos instalado aparecerá esta ventana, es muy importante recordar la contraseña, la mía será 1234.

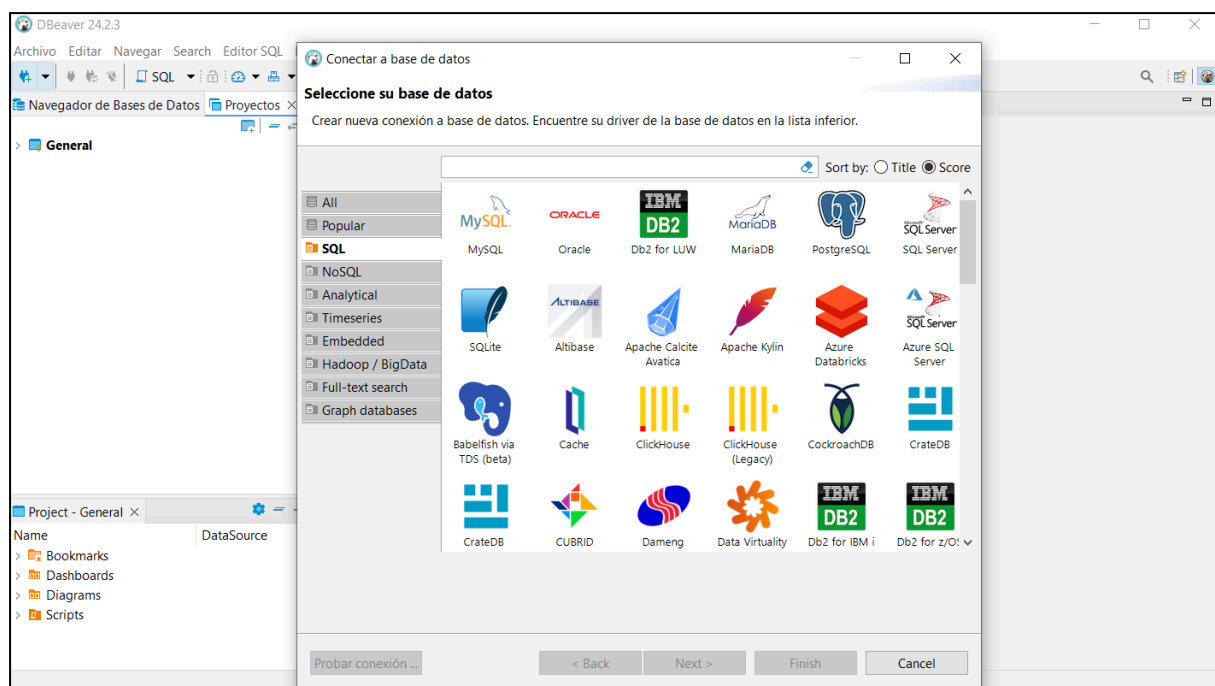
Una vez que tengamos instaladas esas dos cosas vamos a empezar a hacer la conexión:



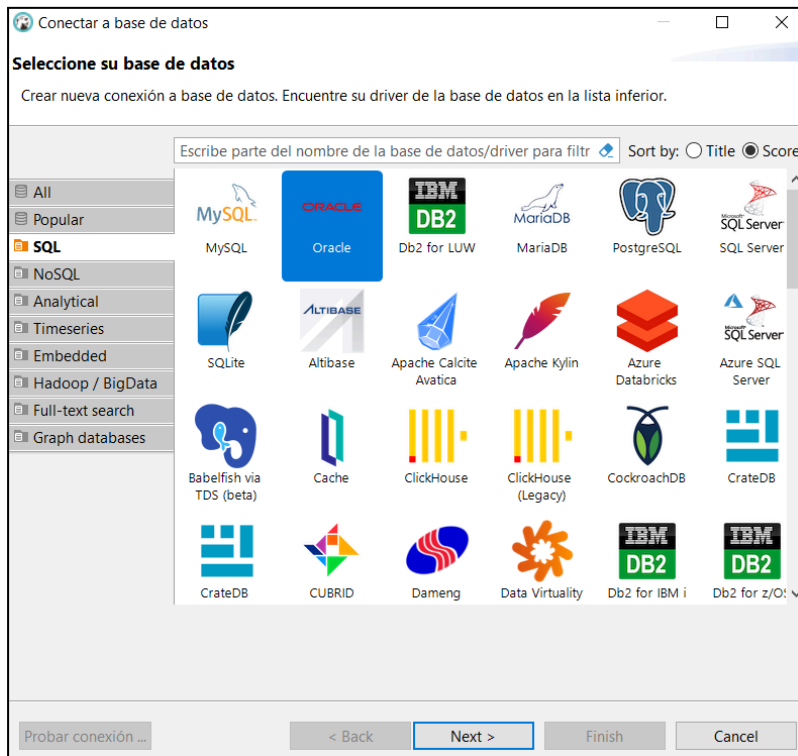
Esta es la primera ventana que te aparece al abrir DBeaver.

**RECUERDA QUE SI TIENES ABIERTO EL XAMPP TIENE QUE ESTAR TOTALMENTE CERRADO.**

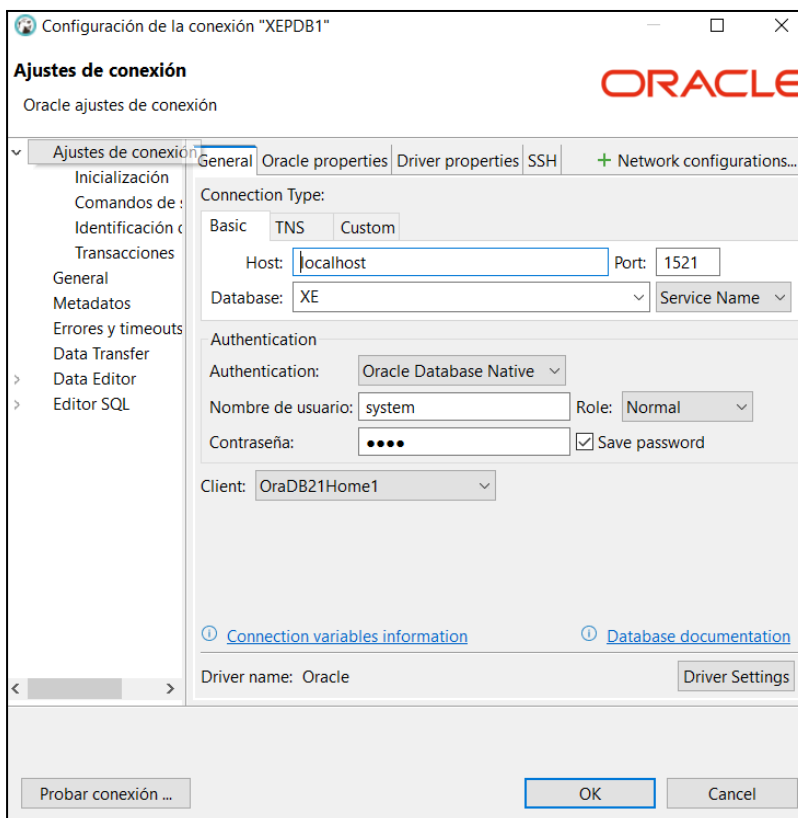
Para hacer la conexión vamos a seguir los siguientes pasos:



Arriba a la izquierda nos sale un símbolo de un enchufe con un más verde pequeño, es para abrir una conexión. Al darle aparecerá esta ventana con los tipos de base de datos que hay, buscamos la de oracle.



Pulsamos a oracle y a next:



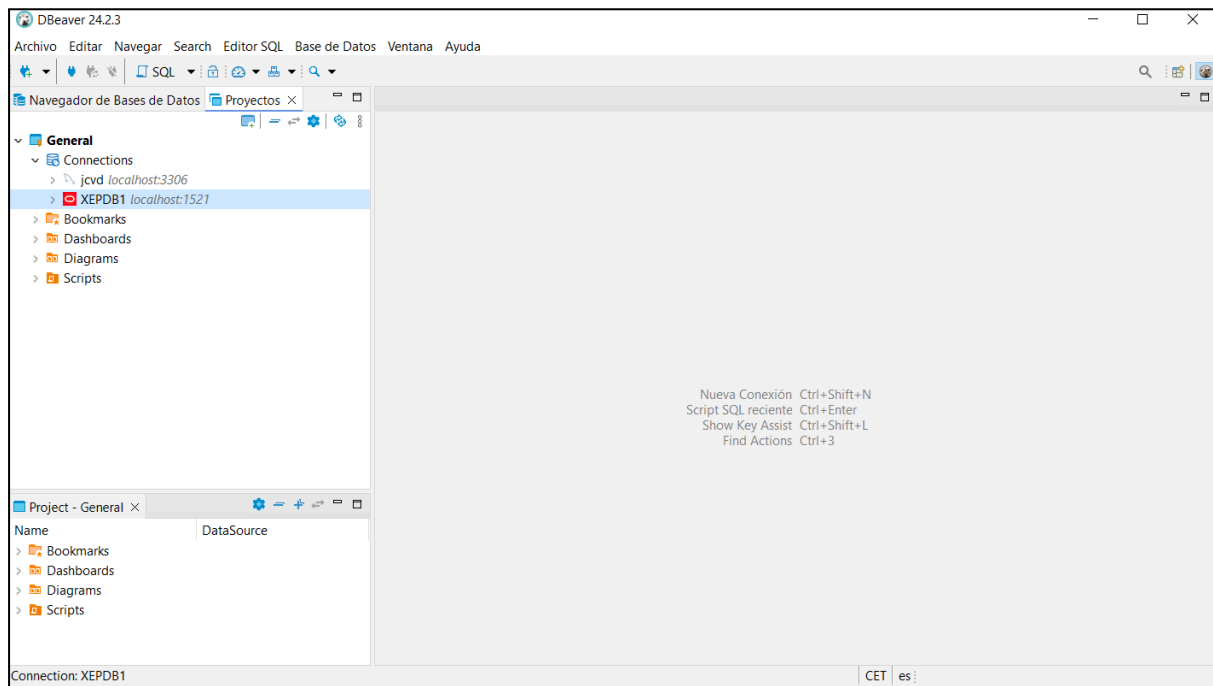
Al darle a next aparecerá esta ventana, tenemos que modificar Database, te aparecerá ORCL de Oracle, lo cambiamos por XE.

Añadiremos una contraseña de la cual nos tendremos que acordar, la mía será 1234 (**Es la misma contraseña que habéis puesto cuando habéis instalado Oracle.**)

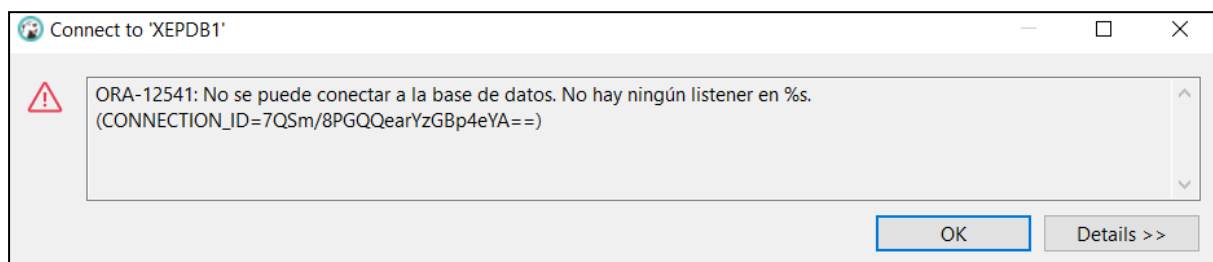
Le damos a Finish una vez que lo tengamos completo.

## ERROR:

Si te das cuenta, en el localhost no aparece que está conectado:



Le clickeamos dos veces:



Si también te aparece este error vamos a seguir los siguientes pasos para solucionarlo:

Vamos a abrir el cmd como administrador:

Vamos a ejecutar lsnrctl status:

```
C:\WINDOWS\system32>lsnrctl status

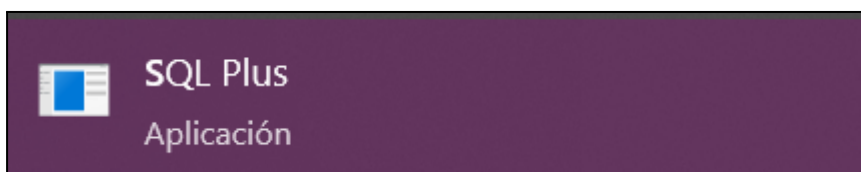
LSNRCTL for 64-bit Windows: Version 21.0.0.0.0 - Production on 30-ENE-2025 18:16:54

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Conectándose a (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host.docker.internal)(PORT=1521)))
TNS-12535: TNS:timeout de la operación
TNS-12560: TNS:error del adaptador de protocolo
TNS-00505: Timeout de la operación
64-bit Windows Error: 60: Unknown error
Conectándose a (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
ESTADO del LISTENER
-----
Alias                LISTENER
Versión              TNSLSNR for 64-bit Windows: Version 21.0.0.0.0 - Production
Fecha de Inicio      28-ENE-2025 09:13:44
Tiempo Actividad     2 días 9 hr. 3 min. 34 seg.
Nivel de Rastreo     off
Seguridad            ON: Local OS Authentication
SNMP                OFF
Servicio por Defecto XE
Parámetros del Listener C:\app\Laura\product\21c\homes\OraDB21Home1\network\admin\listener.ora
Log del Listener      C:\app\Laura\product\21c\diag\tnslsnr\LAPTOP-FHQR2BII\listener>alert\log.xml
Recibiendo Resumen de Puntos Finales...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=LAPTOP-FHQR2BII)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROC1521ipc)))
Resumen de Servicios...
El servicio "CLRExtProc" tiene 1 instancia(s).
  La instancia "CLRExtProc", con estado UNKNOWN, tiene 1 manejador(es) para este servicio...
El comando ha terminado correctamente

C:\WINDOWS\system32>
```

Si no vemos XEPDB1 continuamos con los siguientes pasos:



Iniciamos SQL Plus:

```
SQL*Plus: Release 21.0.0.0.0 - Production on Jue Ene 30 18:20:39 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Introduzca el nombre de usuario: system
Introduzca la contraseña:
Hora de última Conexión Correcta: Jue Ene 30 2025 09:28:10 +01:00

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> _
```

Nos pide el nombre de usuario y la contraseña, el nombre de usuario lo hemos dejado como system y la contraseña 1234.

**Pon la siguiente secuencia:**

```
SELECT NAME FROM V$SERVICES;
```

**Si XEPDB1 no aparece**, significa que no está abierto.

```
ALTER PLUGGABLE DATABASE XEPDB1 OPEN;
```

```
SELECT NAME, OPEN_MODE FROM V$PDBS;
```

**Debe aparecer así:**

```
NAME      OPEN_MODE
-----
PDB$SEED  READ ONLY
XEPDB1    READ WRITE
```

**Ahora vamos a registrar la base de datos en el listener:**

```
ALTER SYSTEM REGISTER;
```

**Reiniciamos el listener (CMD):**

```
lsnrctl stop
lsnrctl start
```

**Verificamos otra vez el listener:**

```
lsnrctl status
```

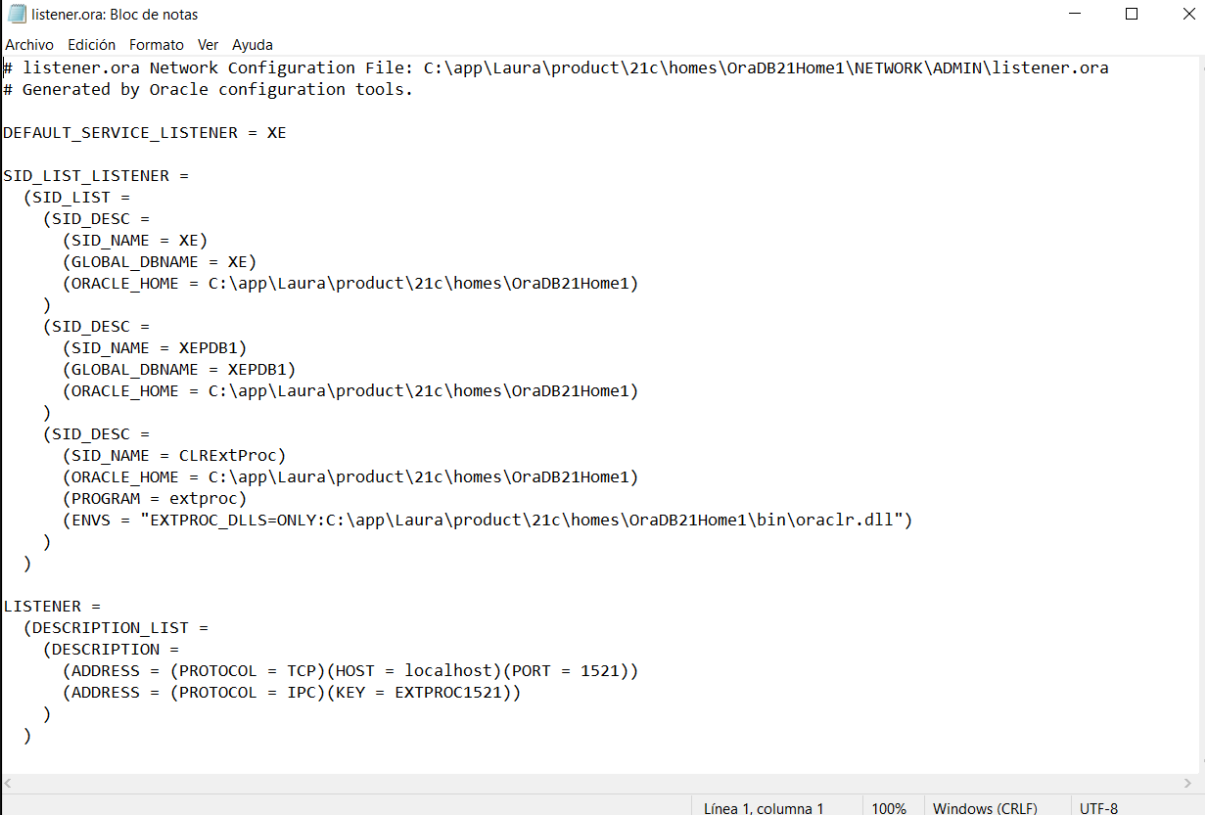
**Si hacemos todos los pasos correctamente debería de funcionar, en el caso de que no lo hiciera debemos modificar el siguiente archivo (CMD):**

```
listener.ora
```

**Pon la ubicación de tu archivo listener.ora en el cmd:**

```
C:\app\tu_usuario\product\21c\homes\OraDB21Home1\network\admin\listener.ora
```

**Se te abrirá automáticamente:**



```
listener.ora: Bloc de notas
Archivo Edición Formato Ver Ayuda
# listener.ora Network Configuration File: C:\app\Laura\product\21c\homes\OraDB21Home1\NETWORK\ADMIN\listener.ora
# Generated by Oracle configuration tools.

DEFAULT_SERVICE_LISTENER = XE

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = XE)
      (GLOBAL_DBNAME = XE)
      (ORACLE_HOME = C:\app\Laura\product\21c\homes\OraDB21Home1)
    )
    (SID_DESC =
      (SID_NAME = XEPDB1)
      (GLOBAL_DBNAME = XEPDB1)
      (ORACLE_HOME = C:\app\Laura\product\21c\homes\OraDB21Home1)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\app\Laura\product\21c\homes\OraDB21Home1)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=ONLY:C:\app\Laura\product\21c\homes\OraDB21Home1\bin\oraclr.dll")
    )
  )
)

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
)
```

Tiene que aparecer así, si no lo tienes así, modificarlo y guardarlo.  
Lo he corregido para que pongas XE o XEPDB1 funcione.

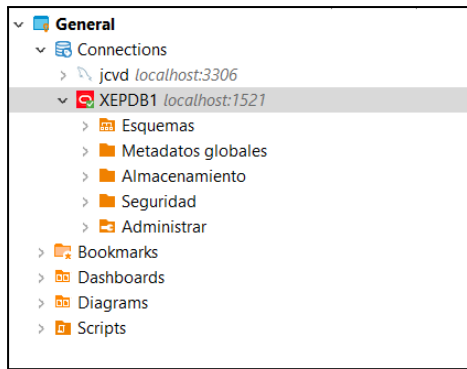
**Reiniciamos el listener (CMD):**

```
lsnrctl stop
lsnrctl start
```

**Verificamos otra vez el listener:**

```
lsnrctl status
```

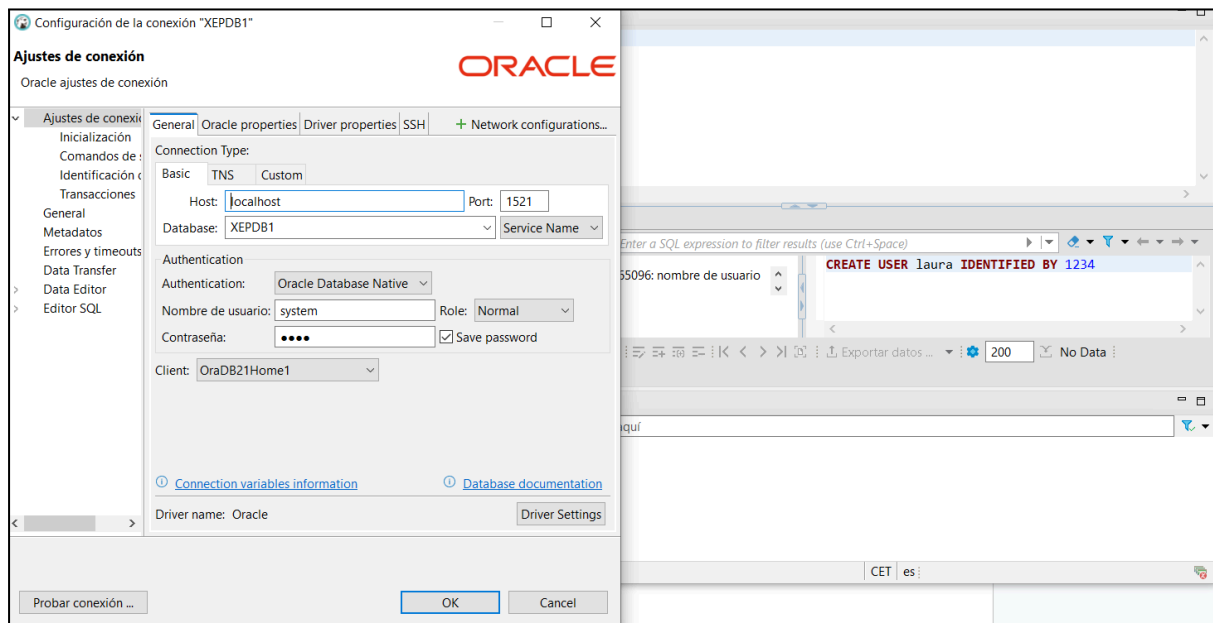




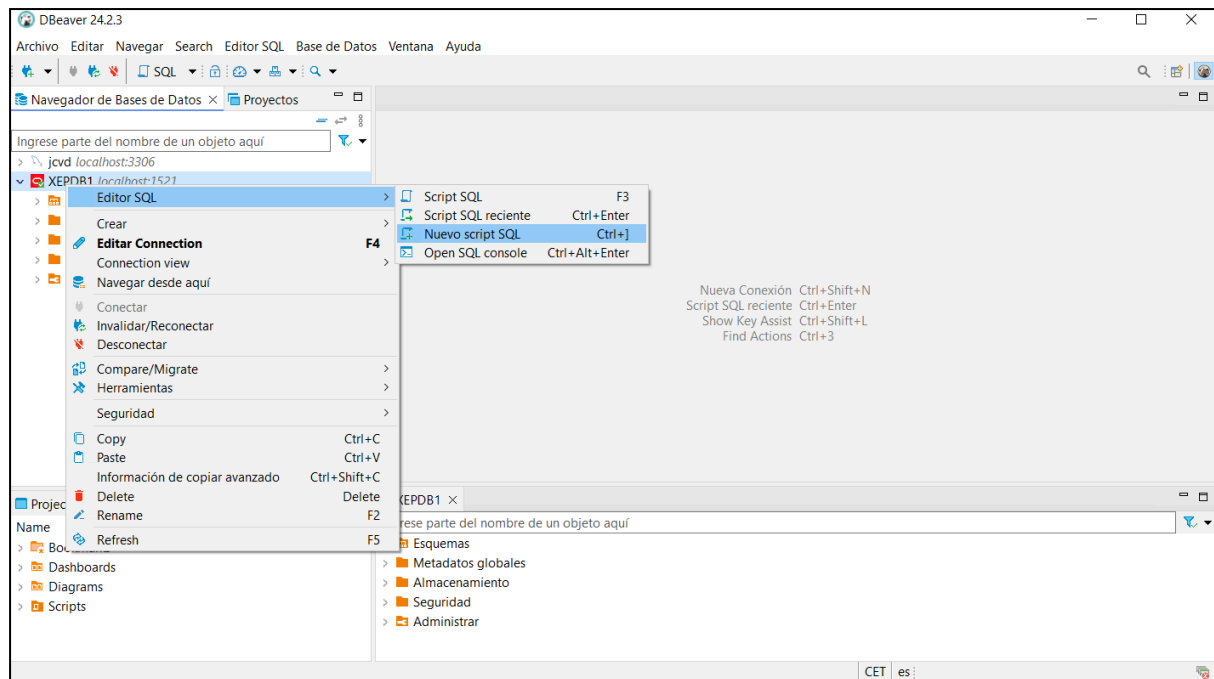
Una vez tengamos el tick verde dignificará que se conecta a la base de datos de Oracle y el error se ha subsanado.

### Vamos a crear un usuario:

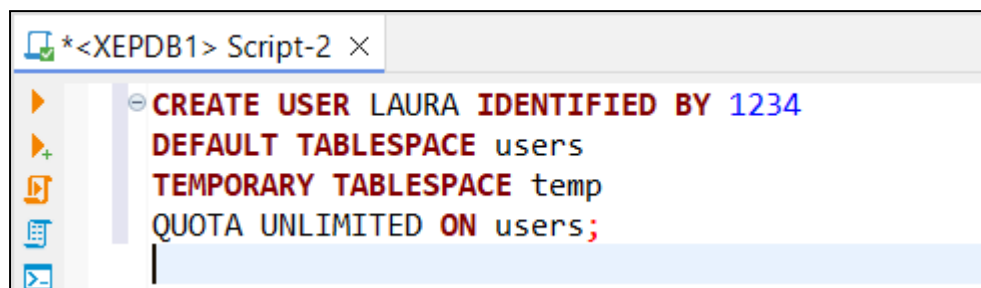
Para crear un usuario nos metemos en la configuración de la base de datos y ponemos XEPDB1.



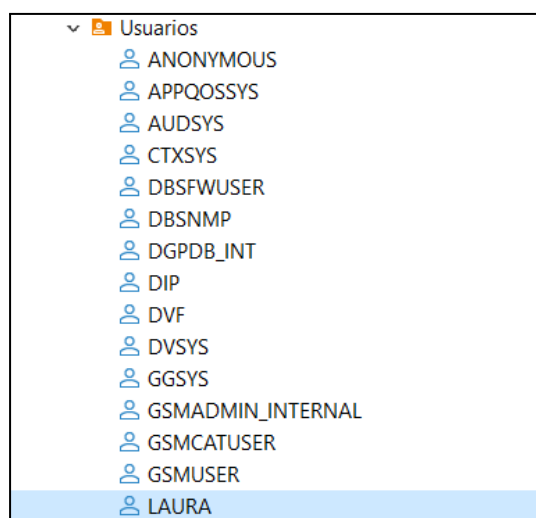
**Con XE NO podrás crear usuarios.**



Lo vamos a hacer con secuencias SQL:



Oracle requiere que el nombre de usuario esté en mayúsculas o entre “”.



En la carpeta usuarios aparece nuestro usuario.

```
GRANT CREATE SESSION TO LAURA;  
GRANT CONNECT TO LAURA;
```

Y estos comandos son para que el usuario pueda iniciar sesión.

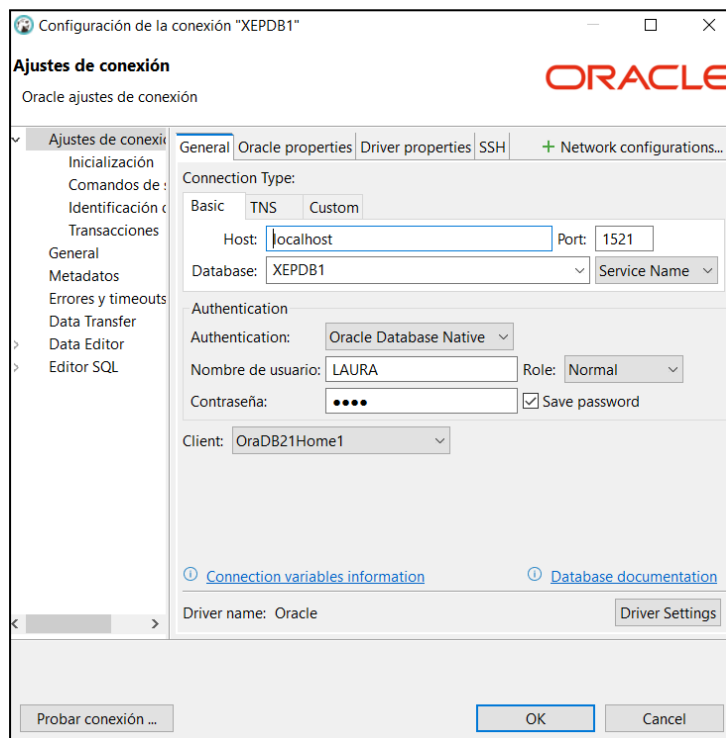
```
GRANT RESOURCE TO LAURA;  
GRANT UNLIMITED TABLESPACE TO LAURA;  
GRANT CREATE TABLE TO LAURA;  
GRANT CREATE VIEW TO LAURA;  
GRANT CREATE PROCEDURE TO LAURA;  
GRANT CREATE SEQUENCE TO LAURA;  
GRANT CREATE SYNONYM TO LAURA;  
GRANT CREATE TRIGGER TO LAURA;  
GRANT CREATE TYPE TO LAURA;  
GRANT CREATE MATERIALIZED VIEW TO LAURA;
```

Permisos sobre objetos y estructuras.

```
GRANT DBA TO LAURA;
```

Este comando es para darle permisos generales como a el usuario system.

Después de tener todos los permisos ya nos podemos conectar a Oracle.



Ahora vamos a crear un proyecto en Apache NetBeans para probar la conexión:

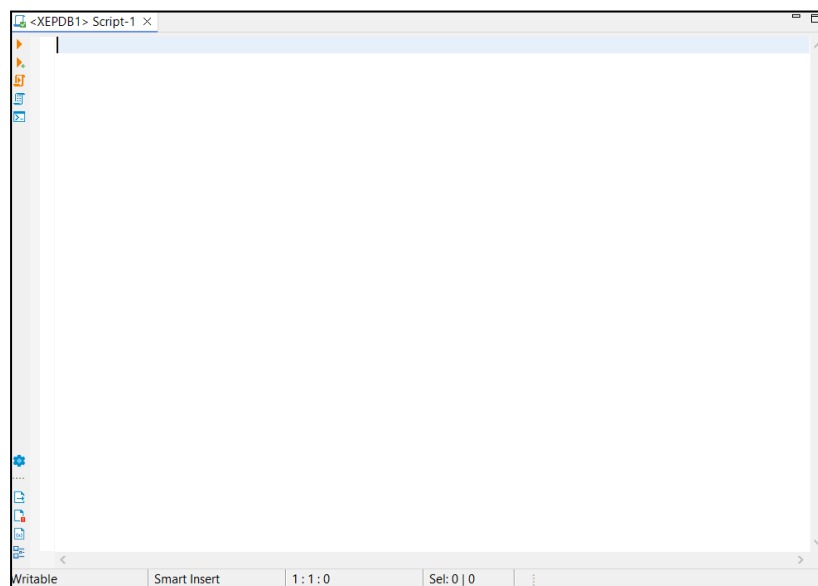
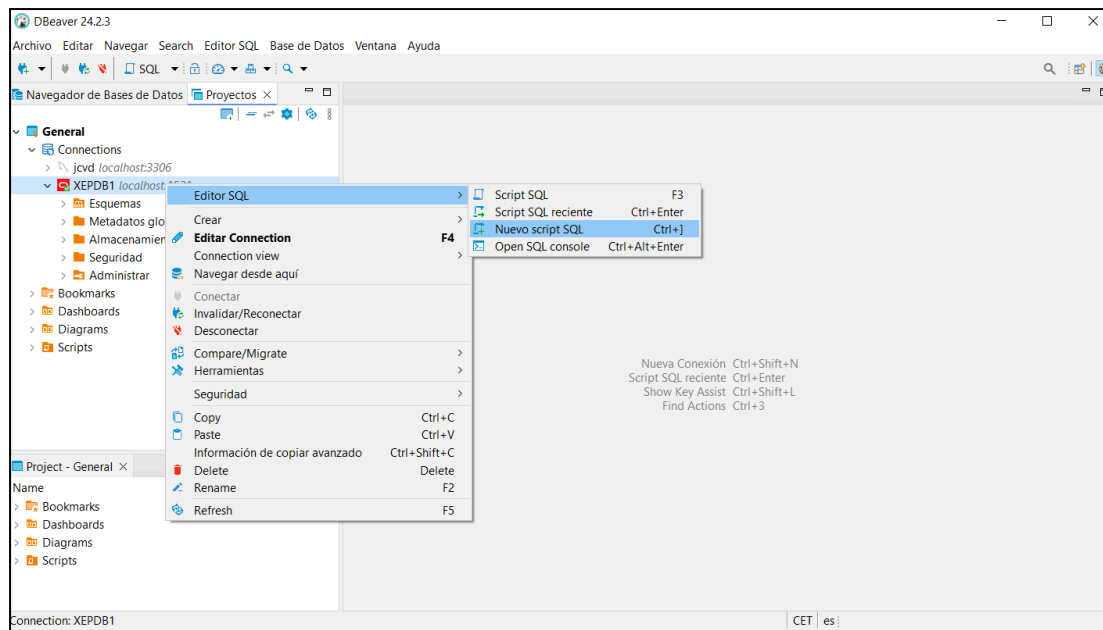
**En una nueva clase ponemos el siguiente código (USUARIO SYSTEM CONEXIÓN XE):**

```
public class PruebaAccesoOracle1 {  
  
    public static void main(String[] args) {  
        try {  
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            Connection con =  
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system", "1234");  
            System.out.println("Conexion OK");  
            //CONSULTAS O ACTUALIZACIONES DE LA BDD  
            con.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Si en el terminal aparece así está bien configurado:

```
--- exec:3.1.0:exec (default-cli) @ PruebaAccesoOracle1 ---  
Conexion OK  
-----  
BUILD SUCCESS  
-----  
Total time: 9.958 s  
Finished at: 2025-01-30T19:58:50+01:00  
-----
```

Ahora vamos a hacer secuencias SQL para hacer tablas en DBeaver:



Se nos abriría este script para hacer secuencias, en nuestro caso será para comprobar si se han hecho las tablas:

**Hacemos otra clase y ponemos código (USUARIO CREADO ANTERIORMENTE Y CONEXIÓN XEPDB1):**

```
package com.mycompany.pruebaaccesooracle1;

import java.sql.*;

public class OracleEjercicio {
    public static void main(String[] args) {
        // Configuración de conexión a la base de datos Oracle
        String url = "jdbc:oracle:thin:@localhost:1521/XEPDB1";
        String user = "LAURA";
        String password = "1234";

        try (Connection con = DriverManager.getConnection(url, user, password)) {
            System.out.println("Conexión OK");

            con.setAutoCommit(false); //Desactivar auto-commit para mayor control de
            transacciones

            borrarTablasYTipos(con);
            crearTipoEstudiante(con);
            crearTablaMisAlumnos(con);
            crearTablaAdmitidos(con);

            insertarEstudiante(con, "Laura", "123456789");
            insertarAdmitido(con, "Laura");

            listarEstudiantes(con);
            listarAdmitidos(con);

            con.commit(); //Confirmar todas las transacciones realizadas
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    // Borra todas las tablas y tipos antes de crearlos nuevamente
    private static void borrarTablasYTipos(Connection con) throws SQLException {
        System.out.println("Eliminando tablas y tipos existentes...");

        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("DROP TABLE MISALUMNOS CASCADE CONSTRAINTS
PURGE");
            System.out.println("Tabla MISALUMNOS eliminada.");
        } catch (SQLException e) {
            System.out.println("No se pudo eliminar MISALUMNOS: " + e.getMessage());
        }
    }
}
```

```

        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("DROP TABLE ADMITIDOS CASCADE CONSTRAINTS
PURGE");
            System.out.println("Tabla ADMITIDOS eliminada.");
        } catch (SQLException e) {
            System.out.println("No se pudo eliminar ADMITIDOS: " + e.getMessage());
        }

        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("DROP SEQUENCE SEQ_ADMITIDOS");
            System.out.println("Secuencia SEQ_ADMITIDOS eliminada.");
        } catch (SQLException e) {
            System.out.println("No se pudo eliminar SEQ_ADMITIDOS: " + e.getMessage());
        }

        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("DROP TYPE ESTUDIANTE FORCE");
            System.out.println("Tipo ESTUDIANTE eliminado.");
        } catch (SQLException e) {
            System.out.println("No se pudo eliminar el tipo ESTUDIANTE: " + e.getMessage());
        }
    }

    // Crea el tipo ESTUDIANTE
    private static void crearTipoEstudiante(Connection con) throws SQLException {
        String sql = "CREATE OR REPLACE TYPE ESTUDIANTE AS OBJECT (" +
            "NOMBRE VARCHAR2(100), " +
            "TELEFONO VARCHAR2(20))";
        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate(sql);
            System.out.println("Tipo ESTUDIANTE creado.");
        }
    }

    // Crea la tabla MISALUMNOS basada en ESTUDIANTE
    private static void crearTablaMisAlumnos(Connection con) throws SQLException {
        String sql = "CREATE TABLE MISALUMNOS OF ESTUDIANTE";
        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate(sql);
            System.out.println("Tabla MISALUMNOS creada.");
        }
    }

    // Crea la tabla ADMITIDOS sin tipo ESTUDIANTE
    private static void crearTablaAdmitidos(Connection con) throws SQLException {
        String sqlCreateTable = "CREATE TABLE ADMITIDOS (" +
            "ID NUMBER PRIMARY KEY, " +

```

```

        "NOMBRE VARCHAR2(100), " +
        "FECHA_ADMISION DATE DEFAULT SYSDATE)";
try (Statement stmt = con.createStatement()) {
    stmt.executeUpdate(sqlCreateTable);
    System.out.println("Tabla ADMITIDOS creada.");
}

String sqlCreateSequence = "CREATE SEQUENCE SEQ_ADMITIDOS START WITH 1
INCREMENT BY 1 NOCACHE";
try (Statement stmt = con.createStatement()) {
    stmt.executeUpdate(sqlCreateSequence);
    System.out.println("Secuencia SEQ_ADMITIDOS creada.");
}

String sqlCreateTrigger = "CREATE OR REPLACE TRIGGER TRG_ADMITIDOS_ID "
+
        "BEFORE INSERT ON ADMITIDOS " +
        "FOR EACH ROW " +
        "BEGIN " +
        "SELECT SEQ_ADMITIDOS.NEXTVAL INTO :NEW.ID FROM DUAL; "
+
        "END;";
try (Statement stmt = con.createStatement()) {
    stmt.executeUpdate(sqlCreateTrigger);
    System.out.println("Trigger TRG_ADMITIDOS_ID creado.");
}
}

// Inserta un estudiante en MISALUMNOS
private static void insertarEstudiante(Connection con, String nombre, String telefono)
throws SQLException {
    Struct estudiante = con.createStruct("ESTUDIANTE", new Object[]{nombre, telefono});
    String sql = "INSERT INTO MISALUMNOS VALUES (?)";

    try (PreparedStatement pstmt = con.prepareStatement(sql)) {
        pstmt.setObject(1, estudiante);
        pstmt.executeUpdate();
        System.out.println("Estudiante insertado en MISALUMNOS.");
    }
}

// Inserta un estudiante en ADMITIDOS sin usar ESTUDIANTE
private static void insertarAdmitido(Connection con, String nombre) throws SQLException
{
    String sql = "INSERT INTO ADMITIDOS (NOMBRE) VALUES (?)";

    try (PreparedStatement pstmt = con.prepareStatement(sql)) {
        pstmt.setString(1, nombre);
    }
}

```



```

        pstmt.executeUpdate();
        System.out.println("Estudiante admitido en ADMITIDOS.");
    }
}

// Lista todos los estudiantes en MISALUMNOS
private static void listarEstudiantes(Connection con) throws SQLException {
    String sql = "SELECT VALUE(m) FROM MISALUMNOS m";
    try (Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
        while (rs.next()) {
            Struct estudiante = (Struct) rs.getObject(1);
            Object[] atributos = estudiante.getAttributes();
            System.out.println("Nombre: " + atributos[0] + ", Teléfono: " + atributos[1]);
        }
    }
}

// Lista todos los estudiantes admitidos desde ADMITIDOS
private static void listarAdmitidos(Connection con) throws SQLException {
    String sql = "SELECT NOMBRE FROM ADMITIDOS";
    try (Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
        while (rs.next()) {
            System.out.println("Admitido: " + rs.getString("NOMBRE"));
        }
    }
}
}

```

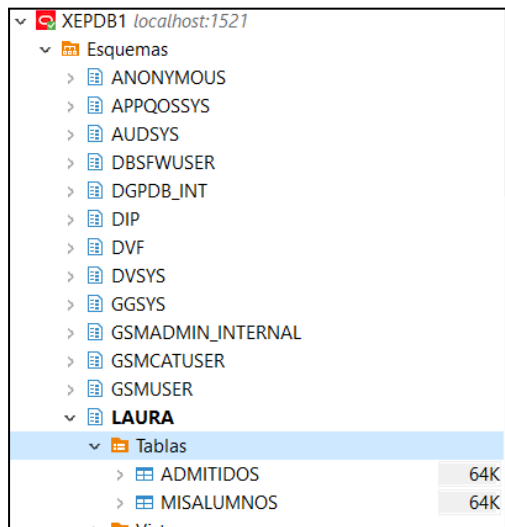
Si lo ejecutamos tendría que salir algo así:

```

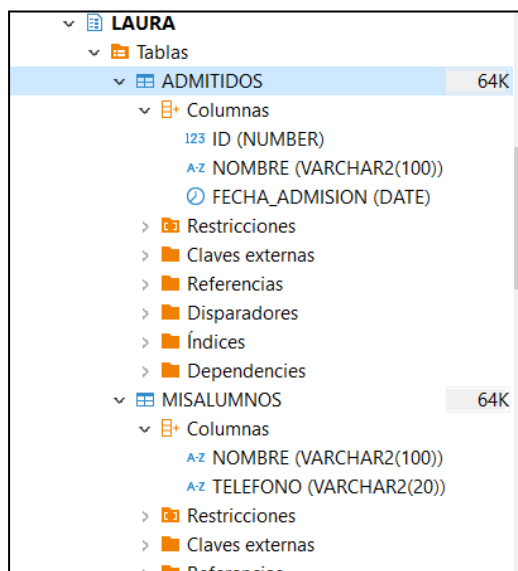
--- exec:3.1.0:exec (default-cli) @ PruebaAccesoOracle1 ---
Conexión OK
Eliminando tablas y tipos existentes...
Tabla MISALUMNOS eliminada.
Tabla ADMITIDOS eliminada.
Secuencia SEQ_ADMITIDOS eliminada.
Tipo ESTUDIANTE eliminado.
Tipo ESTUDIANTE creado.
Tabla MISALUMNOS creada.
Tabla ADMITIDOS creada.
Secuencia SEQ_ADMITIDOS creada.
Trigger TRG_ADMITIDOS_ID creado.
Estudiante insertado en MISALUMNOS.
Estudiante admitido en ADMITIDOS.
Nombre: Laura, Teléfono: 123456789
Admitido: Laura
-----
BUILD SUCCESS
-----
Total time: 2.324 s
Finished at: 2025-01-30T20:12:16+01:00
-----

```

Ahora vamos a ver en DBeaver las tablas que se han creado:

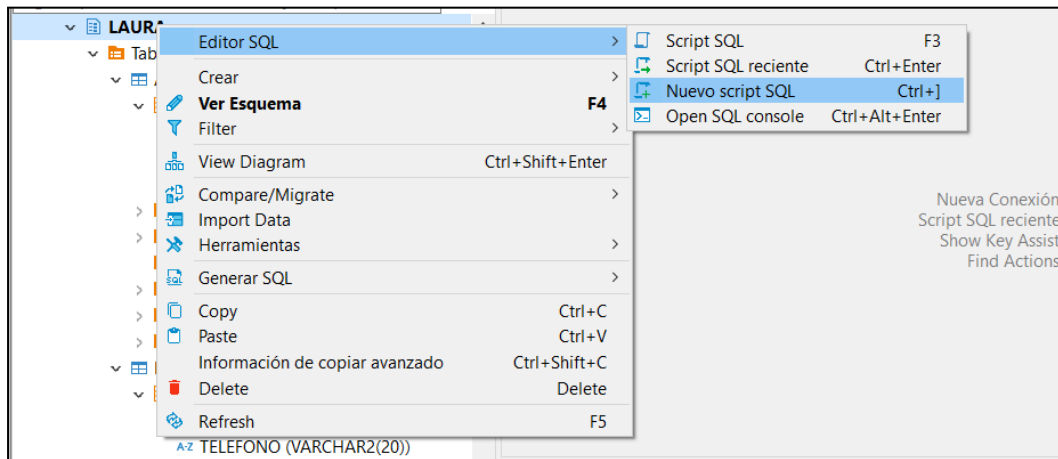


El usuario aparece en negrita porque estamos conectados a él.

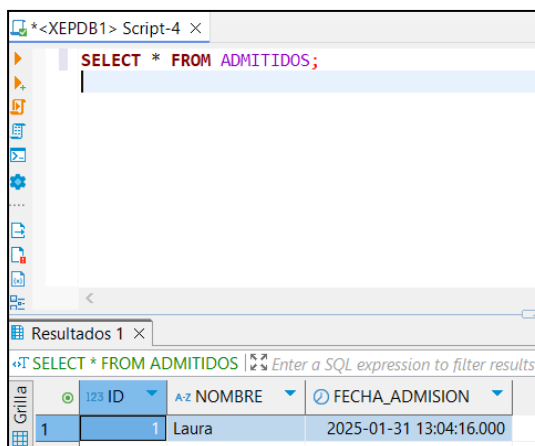
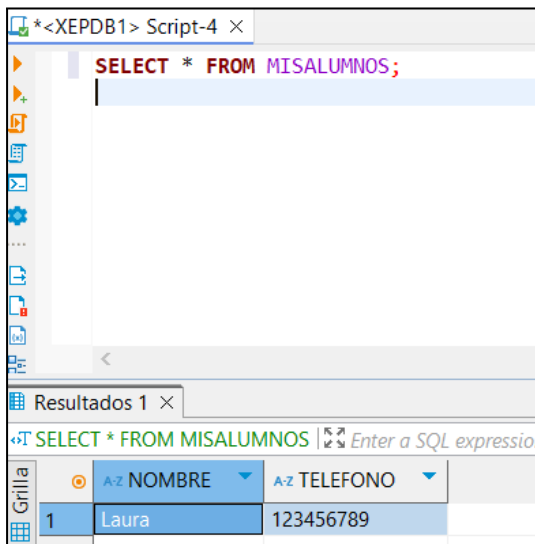


Desde aquí podemos ver las columnas.

Por último vamos a ver el contenido de cada tabla:



Abrimos un script y ponemos este comando.



Solo sale un dato porque en el código solo hemos añadido uno.

FIN DEL TUTORIAL. :)