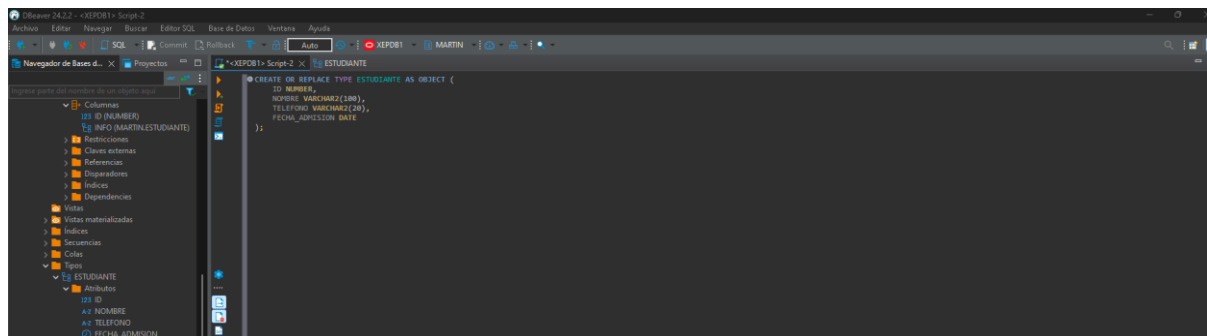


EJERCICIOS 4.1. ORACLE En el código de la práctica 4.1 añade el código para:

1) Crea una tabla misAlumnos de tipo estudiante (de la forma en que se creó la tabla contactos, pero usando el tipo ESTUDIANTE)



CREATE OR REPLACE TYPE ESTUDIANTE AS OBJECT (

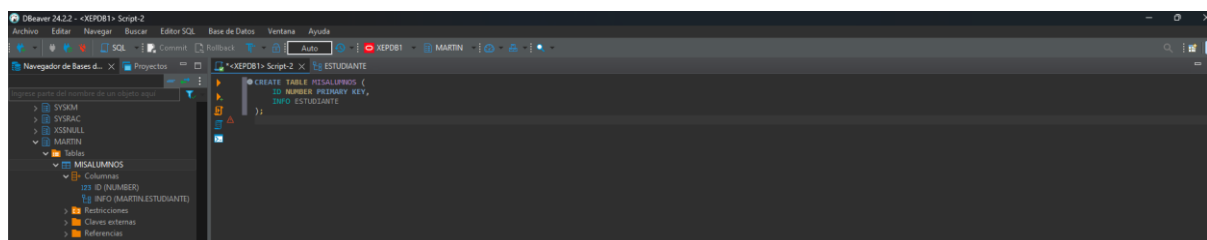
ID NUMBER,

NOMBRE VARCHAR2(100),

TELEFONO VARCHAR2(20),

FECHA_ADMISION DATE

);



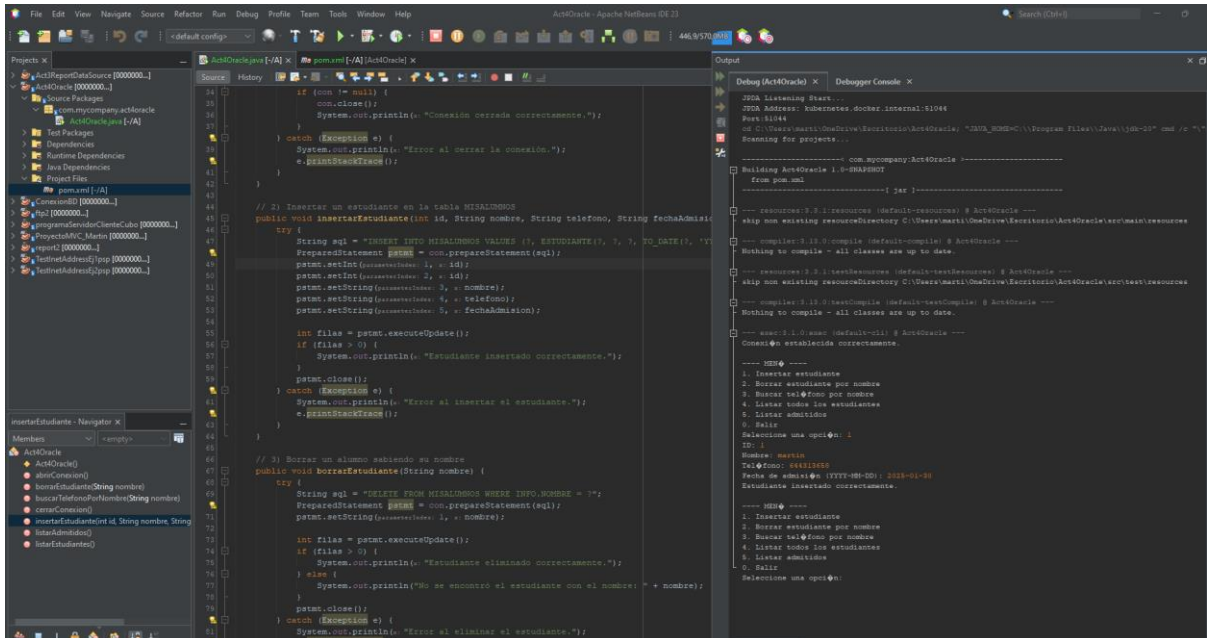
CREATE TABLE MISALUMNOS (

ID NUMBER PRIMARY KEY,

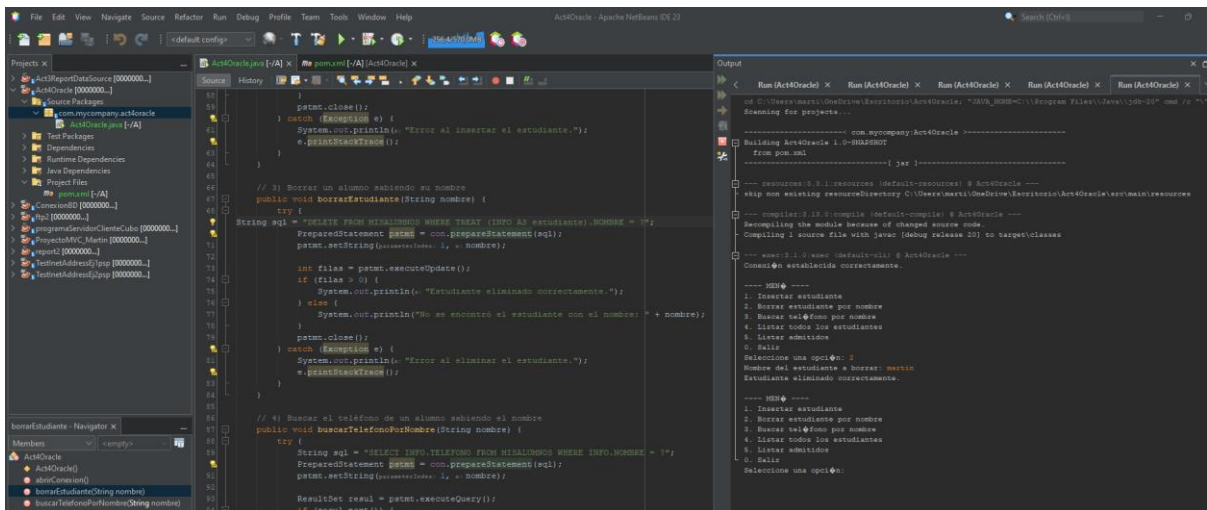
INFO ESTUDIANTE

);

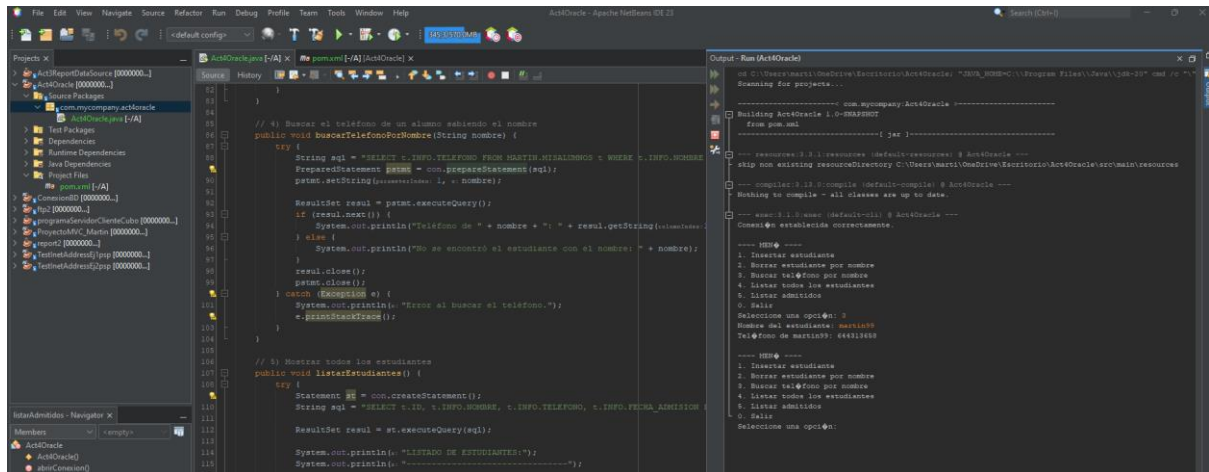
2) Insertar un estudiante en la tabla MISALUMNOS



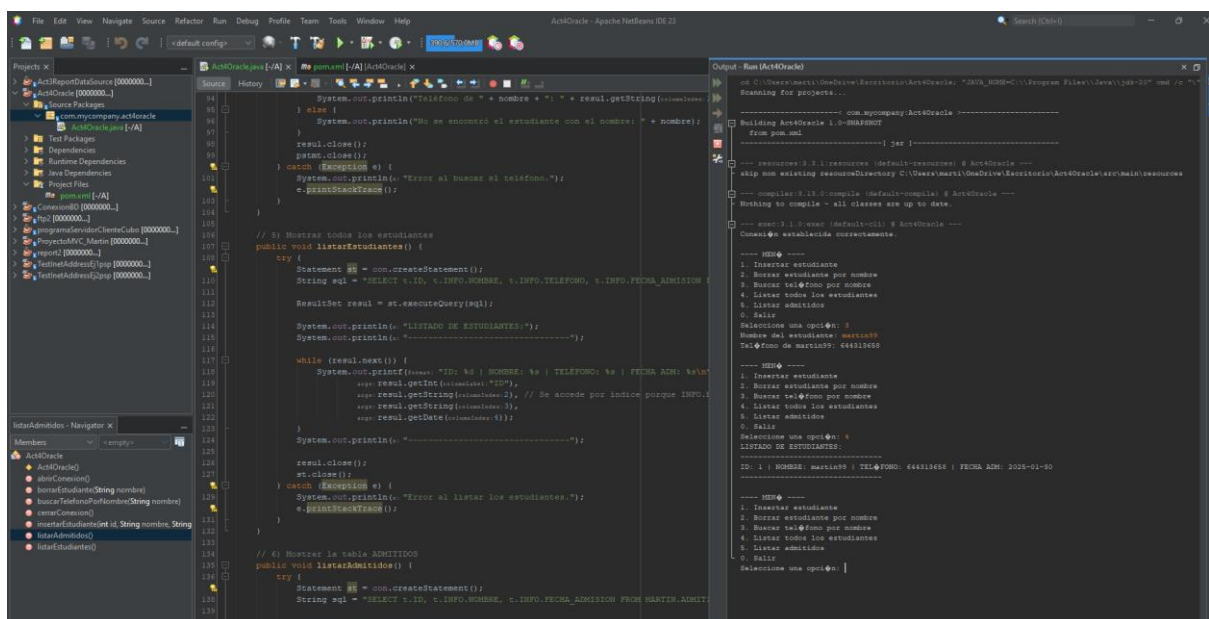
3) Borrar un alumno sabiendo su nombre. Suponemos que no hay dos alumnos con el mismo nombre:



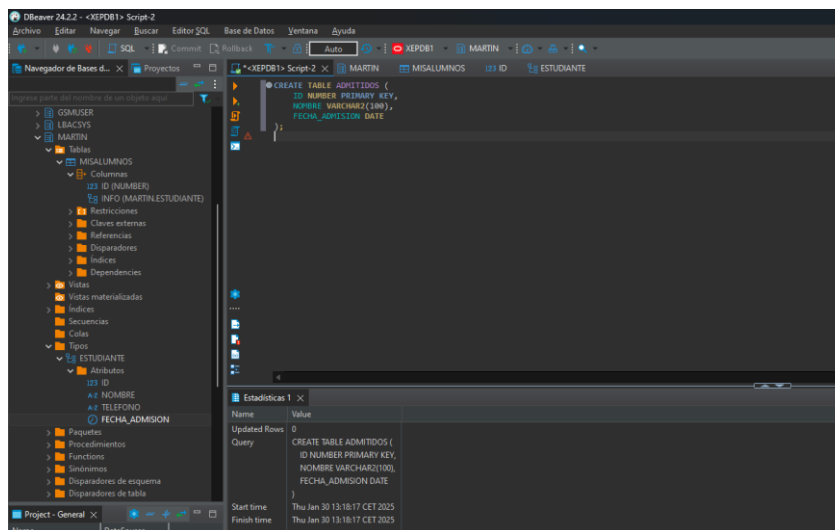
4) Buscar el teléfono de un alumno sabiendo el nombre



5) Mostrar un listado con toda la información de todos los alumnos (se puede hacer con un select similar a los que hay en la presentación powerpoint, o usando STRUCT)



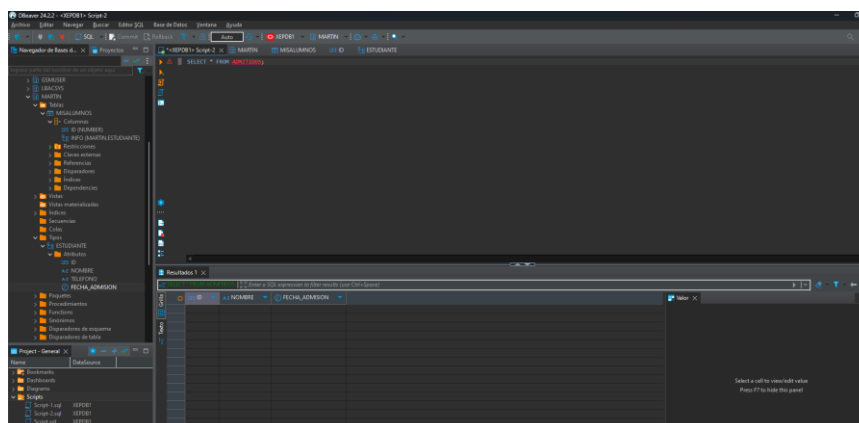
6) Mostrar toda la información de la TABLA ADMITIDOS (se puede hacer con un select similar al que hay en la presentación powerpoint, o usando STRUCT).

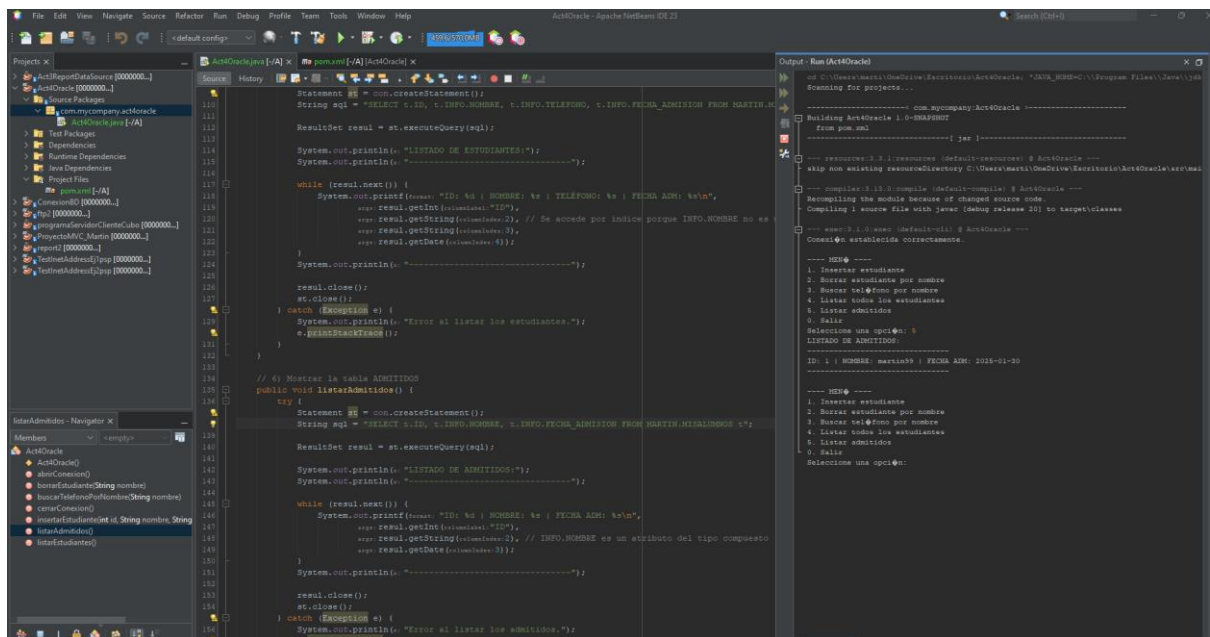
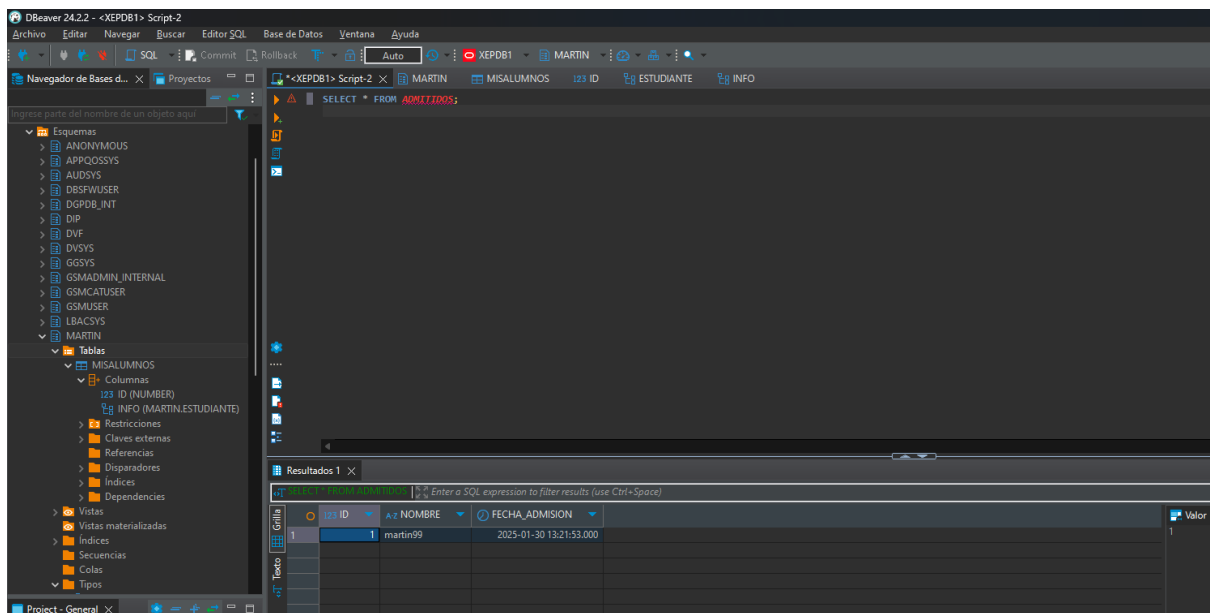


```
CREATE TABLE ADMITIDOS (  
  ID NUMBER PRIMARY KEY,  
  NOMBRE VARCHAR2(100),  
  FECHA_ADMISION DATE  
);
```

```
SELECT * FROM ADMITIDOS;
```

```
INSERT INTO ADMITIDOS (ID, NOMBRE, FECHA_ADMISION)  
VALUES (1, 'martin99', SYSDATE);
```





CÓDIGO JAVA:

```
package com.mycompany.act4oracle;

import java.sql.*; import java.util.Scanner;

/** *

@author martin

*/ public class Act4Oracle {

private Connection con;

// Método para abrir la conexión
public void abrirConexion() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(
            "jdbc:oracle:thin:@//localhost:1521/XEPDB1",
            "MARTIN", // Usuario de la BD
            "MARRTIN17m" // Contraseña de la BD
        );
        System.out.println("Conexión establecida correctamente.");
    } catch (Exception e) {
        System.out.println("Error al conectar con la base de datos.");
        e.printStackTrace();
    }
}

// Método para cerrar la conexión
public void cerrarConexion() {
    try {
        if (con != null) {
            con.close();
            System.out.println("Conexión cerrada correctamente.");
        }
    } catch (Exception e) {
        System.out.println("Error al cerrar la conexión.");
        e.printStackTrace();
    }
}
```

```
}
```

```
// 2) Insertar un estudiante en la tabla MISALUMNOS
```

```
public void insertarEstudiante(int id, String nombre, String telefono, String  
fechaAdmision) {
```

```
    try {
```

```
        String sql = "INSERT INTO MISALUMNOS VALUES (?, ESTUDIANTE(?, ?, ?,  
TO_DATE(?, 'YYYY-MM-DD'))))";
```

```
        PreparedStatement pstmt = con.prepareStatement(sql);
```

```
        pstmt.setInt(1, id);
```

```
        pstmt.setInt(2, id);
```

```
        pstmt.setString(3, nombre);
```

```
        pstmt.setString(4, telefono);
```

```
        pstmt.setString(5, fechaAdmision);
```

```
        int filas = pstmt.executeUpdate();
```

```
        if (filas > 0) {
```

```
            System.out.println("Estudiante insertado correctamente.");
```

```
        }
```

```
        pstmt.close();
```

```
    } catch (Exception e) {
```

```
        System.out.println("Error al insertar el estudiante.");
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
// 3) Borrar un alumno sabiendo su nombre
```

```
public void borrarEstudiante(String nombre) {
```

```
    try {
```

```
        String sql = "DELETE FROM MISALUMNOS WHERE TREAT (INFO AS  
estudiante).NOMBRE = ?";
```

```
        PreparedStatement pstmt = con.prepareStatement(sql);
```

```
        pstmt.setString(1, nombre);
```

```
        int filas = pstmt.executeUpdate();
```

```
        if (filas > 0) {
```

```
            System.out.println("Estudiante eliminado correctamente.");
```

```
        } else {
```

```
            System.out.println("No se encontró el estudiante con el nombre: " + nombre);
```

```
        }
```

```
        pstmt.close();
```

```

    } catch (Exception e) {
        System.out.println("Error al eliminar el estudiante.");
        e.printStackTrace();
    }
}

```

// 4) Buscar el teléfono de un alumno sabiendo el nombre

```

public void buscarTelefonoPorNombre(String nombre) {
    try {
        String sql = "SELECT t.INFO.TELEFONO FROM MARTIN.MISALUMNOS t WHERE
t.INFO.NOMBRE = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, nombre);

        ResultSet resul = pstmt.executeQuery();
        if (resul.next()) {
            System.out.println("Teléfono de " + nombre + ": " + resul.getString(1));
        } else {
            System.out.println("No se encontró el estudiante con el nombre: " + nombre);
        }
        resul.close();
        pstmt.close();
    } catch (Exception e) {
        System.out.println("Error al buscar el teléfono.");
        e.printStackTrace();
    }
}

```

// 5) Mostrar todos los estudiantes

```

public void listarEstudiantes() {
    try {
        Statement st = con.createStatement();
        String sql = "SELECT t.ID, t.INFO.NOMBRE, t.INFO.TELEFONO,
t.INFO.FECHA_ADMISION FROM MARTIN.MISALUMNOS t";

        ResultSet resul = st.executeQuery(sql);

        System.out.println("LISTADO DE ESTUDIANTES:");
        System.out.println("-----");

        while (resul.next()) {

```



```

        System.out.printf("ID: %d | NOMBRE: %s | TELÉFONO: %s | FECHA ADM: %s\n",
            resul.getInt("ID"),
            resul.getString(2), // Se accede por índice porque INFO.NOMBRE no es una
columna directa
            resul.getString(3),
            resul.getDate(4));
    }
    System.out.println("-----");

    resul.close();
    st.close();
} catch (Exception e) {
    System.out.println("Error al listar los estudiantes.");
    e.printStackTrace();
}
}

```

// 6) Mostrar la tabla ADMITIDOS

```

public void listarAdmitidos() {
    try {
        Statement st = con.createStatement();
        String sql = "SELECT t.ID, t.INFO.NOMBRE, t.INFO.FECHA_ADMISION FROM
MARTIN.MISALUMNOS t";

        ResultSet resul = st.executeQuery(sql);

        System.out.println("LISTADO DE ADMITIDOS:");
        System.out.println("-----");

        while (resul.next()) {
            System.out.printf("ID: %d | NOMBRE: %s | FECHA ADM: %s\n",
                resul.getInt("ID"),
                resul.getString(2), // INFO.NOMBRE es un atributo del tipo compuesto
                resul.getDate(3));
        }
        System.out.println("-----");

        resul.close();
        st.close();
    } catch (Exception e) {
        System.out.println("Error al listar los admitidos.");
    }
}

```

```
        e.printStackTrace();
    }
}
```

// MAIN: Prueba de los métodos

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Act4Oracle a = new Act4Oracle();
    a.abrirConexion();

    int opcion;
    do {
        System.out.println("\n---- MENÚ ----");
        System.out.println("1. Insertar estudiante");
        System.out.println("2. Borrar estudiante por nombre");
        System.out.println("3. Buscar teléfono por nombre");
        System.out.println("4. Listar todos los estudiantes");
        System.out.println("5. Listar admitidos");
        System.out.println("0. Salir");
        System.out.print("Seleccione una opción: ");
        opcion = scanner.nextInt();
        scanner.nextLine(); // Limpiar buffer

        switch (opcion) {
            case 1:
                System.out.print("ID: ");
                int id = scanner.nextInt();
                scanner.nextLine();
                System.out.print("Nombre: ");
                String nombre = scanner.nextLine();
                System.out.print("Teléfono: ");
                String telefono = scanner.nextLine();
                System.out.print("Fecha de admisión (YYYY-MM-DD): ");
                String fechaAdmision = scanner.nextLine();
                a.insertarEstudiante(id, nombre, telefono, fechaAdmision);
                break;

            case 2:
                System.out.print("Nombre del estudiante a borrar: ");
                String nombreBorrar = scanner.nextLine();
                a.borrarEstudiante(nombreBorrar);
```

```
break;
```

```
case 3:
```

```
    System.out.print("Nombre del estudiante: ");  
    String nombreBuscar = scanner.nextLine();  
    a.buscarTelefonoPorNombre(nombreBuscar);  
    break;
```

```
case 4:
```

```
    a.listarEstudiantes();  
    break;
```

```
case 5:
```

```
    a.listarAdmitidos();  
    break;
```

```
case 0:
```

```
    System.out.println("Saliendo...");  
    break;
```

```
default:
```

```
    System.out.println("Opción inválida, intente de nuevo.");
```

```
}
```

```
} while (opcion != 0);
```

```
a.cerrarConexion();
```

```
scanner.close();
```

```
}
```

```
}
```