

BBDD Oracle SQL en Java Eclipse

Paso 1: Descargar conector

Debemos visitar la página web de la BBDD, que en nuestro caso es Oracle, y más concretamente, la sección de descarga de conectores:






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

Una vez en la web, descargaremos el primer conector en .jar que vemos en la lista, el más actualizado y con mayores certificaciones.

Oracle Database 23c (23.3.0.23.09) JDBC Driver & UCP Downloads - Free Release

Supports Oracle Database versions - 23c, 21c, and 19c Refer to [JDBC-UCP-ReleaseNotes-23c.txt](#) and [Bugs-fixed-in-23c.txt](#)

Note: The Oracle JDBC 23.3 drivers are supported for production use with BaseDB only. The 23.3 release drivers are not supported for production for any other database confi

Name	Download	JDK Supported	Description
Oracle JDBC driver	 ojdbc11.jar	Implements JDBC 4.3 spec and certified with JDK11, JDK17, JDK19, and JDK21	Oracle JDBC driver except classes for NLS support (7,107,784 bytes) - (SHA1: 405bcb8d8dab59f5621
Oracle JDBC Driver	 ojdbc8.jar	Implements JDBC 4.2 spec and certified with JDK8 and JDK11	Oracle JDBC driver except classes for NLS support (6,980,481 bytes) - (SHA1: d36f44a0ed8a07dcff2
Universal Connection Pool (UCP) - ucp11.jar	 ucp11.jar	Certified with JDK11, JDK17, JDK19, and JDK21	Universal Connection Pool (UCP) to be used with (1,526,942 bytes) - (SHA1: 2017015e2282f528a7da
Universal Connection Pool (UCP)	 ucp.jar	Certified with JDK8	Universal Connection Pool (UCP) to be used with (1,485,969 bytes) - (SHA1: 9cdb5a2c3e48b140686
Zipped JDBC driver (ojdbc11.jar)	 ojdbc11-full.tar.gz	Certified with JDK11, JDK17, JDK19, and JDK21	This archive contains ojdbc11.jar , ucp11.jar , Reac JDBC. UCP. RSI Javadoc. and their Readmes. Refe

Paso 2: Crear un nuevo proyecto y pegar el conector

Cread un nuevo proyecto en Eclipse (Java Project).

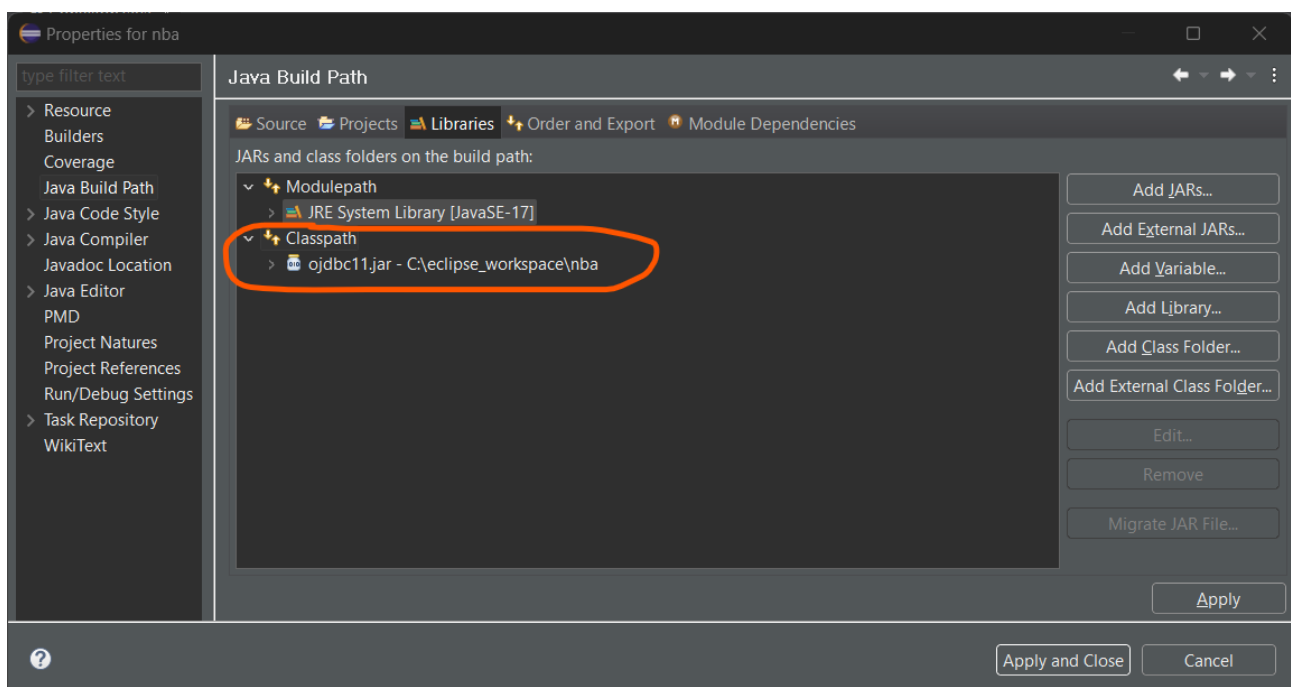
Tras eso, cerramos Eclipse y copiamos el conector que acabamos de descargar a la carpeta del proyecto.

Nombre	Fecha de modificación	Tipo	Tamaño
.settings	02/05/2024 16:36	Carpeta de archivos	
bin	02/05/2024 16:44	Carpeta de archivos	
src	02/05/2024 16:40	Carpeta de archivos	
.classpath	02/05/2024 16:41	Archivo CLASSPATH	1 KB
.project	02/05/2024 16:36	Archivo PROJECT	1 KB
 ojdbc11	02/05/2024 15:14	Executable Jar File	6.942 KB

Ahora, volvemos a abrir Eclipse (esto es para asegurarnos de que Eclipse refresque y detecte el conector, aunque quizás no sea necesario abrirlo y cerrarlo).

Paso 3: Importar el conector al proyecto

1. En Eclipse, abrimos el menú conceptual de nuestro proyecto, y seleccionamos *properties* o *propiedades* (ALT+ENTER).
2. Clic en *Java Build Path* y después en *Libraries*.
3. Clic en *Classpath* (un solo clic para que se resalte).
4. Ahora, se nos resaltarán las opciones del menú derecho, y le haremos clic a *Add External JARs*.
5. Se nos abrirá el Explorador de Archivos de Windows, y debemos buscar y seleccionar nuestro conector.
6. Tras seleccionarlo, debajo de *Classpath*, ahora aparecerá el fichero JAR del conector, de la siguiente manera:



7. Hacemos Clic en *Apply and Close*.

Paso 4: Probar conexión

Si todo ha salido bien, ya tenemos la librería necesaria para conectar una BBDD Oracle SQL a Java. Ahora, en nuestro proyecto, podemos crear la clase en la que accederemos y haremos consultas a la BBDD.

Yo, en mi caso, he hecho la siguiente prueba:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Conexion {
    public static void main(String[] args) throws SQLException {
        // Establecer la conexión con la base de datos
        Connection conex = null;
        try {
            // Cargar el driver JDBC
            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Establecer la conexión
            conex = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "nba", "nba");

            Statement sentencia = conex.createStatement();
            ResultSet resultado = sentencia.executeQuery("SELECT nombre FROM equipos"); //Prueba con una consulta sencilla.

            int cont=0;
            // Procesa los resultados
            while (resultado.next()) {
                String nombre = resultado.getString("nombre");
                System.out.println("Jugador "+(cont+1)+": "+nombre);
                cont++;
            }
            System.out.println("\nSe han encontrado "+cont+" registros.");
            //Cerrar la conexión
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Cabe destacar que “nba”, “nba” son los usuario y contraseña correspondientes al usuario QUE YO TENGO en mi SQL Developer para acceder a MI base de datos (en este caso, de jugadores de la NBA). En vuestro caso esto variará dependiendo del usuario y contraseña.

NOTA → El @localhost:1521 es el puerto más común que se utiliza para añadir una nueva conexión en Oracle SQL Developer, pero esto puede variar dependiendo de nuestras necesidades, aunque lo más probable es que lo tengáis configurado con ese puerto.

Siguiendo la sintaxis aprendida en clase de Programación...

- Usando la conexión recientemente creada, creamos una instancia del objeto Statement para CREAR una sentencia SQL.

- Ahora una instancia de `ResultSet`. Ahí guardamos el resultado de la sentencia SQL que queramos hacer, usando la instancia `Statement` para llamar al método “`executeQuery`”, y poniendo entre paréntesis en formato `String` la consulta en sí.

```
Statement sentencia = conex.createStatement();
ResultSet resultado = sentencia.executeQuery(
    "SELECT nombre FROM equipos"
);
```

NOTA → “`conex`” es el nombre que le he dado yo a la conexión con la Base de Datos.

Usando un bucle `While`, siguiendo el código de la captura, podremos sacar registros de la consulta y manejarlos en consecuencia.

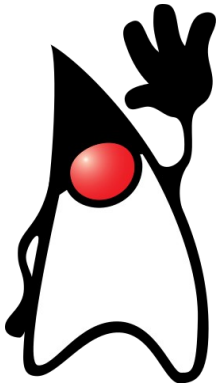
En el caso de mi código, si todo ha funcionado, el resultado será algo similar a esto:

```
Jugador 1: 76ers
Jugador 2: Bobcats
Jugador 3: Bucks
Jugador 4: Bulls
Jugador 5: Cavaliers
Jugador 6: Celtics
Jugador 7: Clippers
Jugador 8: Grizzlies
Jugador 9: Hawks
Jugador 10: Heat
Jugador 11: Hornets
Jugador 12: Jazz
Jugador 13: Kings
Jugador 14: Knicks
Jugador 15: Lakers
Jugador 16: Magic
Jugador 17: Mavericks
Jugador 18: Nets
Jugador 19: Nuggets
Jugador 20: Pacers
Jugador 21: Pistons
Jugador 22: Raptors
Jugador 23: Rockets
Jugador 24: Spurs
Jugador 25: Suns
Jugador 26: Supersonics
Jugador 27: Timberwolves
Jugador 28: Trail Blazers
Jugador 29: Warriors
Jugador 30: Wizards

Se han encontrado 30 registros.
```

	NOMBRE
1	76ers
2	Bobcats
3	Bucks
4	Bulls
5	Cavaliers
6	Celtics
7	Clippers
8	Grizzlies
9	Hawks
10	Heat
11	Hornets
12	Jazz
13	Kings
14	Knicks
15	Lakers
16	Magic
17	Mavericks
18	Nets
19	Nuggets
20	Pacers
21	Pistons
22	Raptors
23	Rockets
24	Spurs
25	Suns
26	Supersonics
27	Timberwolves
28	Trail Blazers
29	Warriors
30	Wizards

Si lo comparamos al resultado de la misma consulta en SQL Developer, vemos que los resultados coinciden y que la conexión ha sido un éxito.



Eso es todo. Buena suerte!

Bibliografía

- <https://stackoverflow.com/>
- <https://chat.openai.com/>
- <https://www.oracle.com/es/>
- <https://www.oracle.com/es/database/technologies/appdev/jdbc-downloads.html>