

Las clases **CreaEmpleados.java** y **CreaDepartamentos.java** permiten crear las tablas de empleados y departamentos e insertar algunos datos en cada una de ellas.

### EJEMPLO 1. [CreaDepartamentos.java]

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
/* clase que ilustra la creación de la tabla 'departamentos' en la base de datos
 * 'empresaz' del servidor 'localhost', así como la inserción de unos cuantos
 * registros de ejemplo. */
public class CreaDepartamentos {
    /* parámetros de conexión con la base de datos MySQL */
    private static final String url = "jdbc:mysql://localhost/empresaz";
    private static final String user = "root";
    private static final String password = "admin";
    /* sentencia de creación de la tabla 'departamentos' */
    private static final String creaDepartamentos =
        "CREATE TABLE departamentos ("
        + "dept_no TINYINT(2) NOT NULL PRIMARY KEY, "
        + "dnombre VARCHAR(15), "
        + "loc VARCHAR(15) "
        + ")";
    /* sentencia de inserción de los registros de ejemplo */
    private static final String insertaDepartamentos =
        "INSERT INTO departamentos VALUES"
        + "(10, 'CONTABILIDAD', 'SEVILLA'), "
        + "(20, 'INVESTIGACIÓN', 'MADRID'), "
        + "(30, 'VENTAS', 'BARCELONA'), "
        + "(40, 'PRODUCCIÓN', 'BILBAO)";

    public static void main(String[] args) {
        /* objeto de conexión */
        Connection conexion = null;
        /* comando para enviar sentencias a través de la conexión */
        Statement sentencia = null;
        try {
            /* carga el driver previamente añadido a las 'Bibliotecas' */
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            /* establece la conexión */
            conexion = DriverManager.getConnection(url, user, password);
            /* crea el comando para enviar sentencias SQL */
            sentencia = conexion.createStatement();
            /* ejecuta la sentencia de creación de la tabla 'departamentos' */
            sentencia.executeUpdate(creaDepartamentos);
            /* mensaje de confirmación */
            System.out.println("Tabla 'departamentos' creada correctamente");
            /* ejecuta la sentencia de inserción de departamentos */
            sentencia.executeUpdate(insertaDepartamentos);
            /* mensaje de confirmación */
            System.out.println("Nuevos departamentos insertados correctamente");
        } catch (Exception ex) {
            /* mensaje de error */
            System.err.println(ex);
        } finally {
            /* si llegó a crearse el comando */
            if (sentencia != null) {
                try {
                    /* trata de liberar recursos */
                    sentencia.close();
                } catch (SQLException ex) {
                }
            }
            /* si llegó a establecerse la conexión */
            if (conexion != null) {
                try {
                    /* trata de liberar recursos */
                    conexion.close();
                }
            }
        }
    }
}
```

```

    } catch (SQLException ex) {
    }
}
}
}
}

```

## EJEMPLO 2. [CreaEmpleados.java]

```

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

/* clase que ilustra la creación de la tabla 'empleados' en la base de datos
 * 'empresaz' del servidor 'localhost', así como la inserción de unos cuantos
 * registros de ejemplo. */
public class CreaEmpleados {
    /* parámetros de conexión con la base de datos MySQL */
    private static final String url = "jdbc:mysql://localhost/empresaz";
    private static final String user = "root";
    private static final String password = "admin";
    /* sentencia de creación de la tabla 'empleados' */
    private static final String creaEmpleados =
        "CREATE TABLE empleados ("
        + "emp_no SMALLINT(4) UNSIGNED NOT NULL PRIMARY KEY, "
        + "apellido VARCHAR(10), "
        + "oficio VARCHAR(10), "
        + "dir SMALLINT, "
        + "fecha_alt DATE, "
        + "salario FLOAT(6,2), "
        + "comision FLOAT(6,2), "
        + "dept_no TINYINT(2) NOT NULL, "
        + "FOREIGN KEY (dept_no) REFERENCES departamentos(dept_no)"
        + ")";

    /* sentencia de inserción de los registros de ejemplo */
    private static final String insertaEmpleados =
        "INSERT INTO empleados VALUES"
        + "(1000, 'AMARO', 'AUXILIAR', 1, 20100102, 940.00, 0.00, 10), "
        + "(2000, 'CASTRO', 'AUXILIAR', 2, 20110102, 890.00, 0.00, 10), "
        + "(3000, 'HERNANDEZ', 'CONTABLE', 3, 20010102, 1890.00, 0.00, 10), "
        + "(4000, 'FERNANDEZ', 'TEC. LAB.', 4, 20080102, 1500.00, 0.00, 20), "
        + "(5000, 'HERMOSILLA', 'DIR. LAB.', 5, 20060102, 2420.00, 0.00, 20), "
        + "(6000, 'DE ANDRES', 'GERENTE', 6, 20010102, 3450.00, 20.00, 40), "
        + "(7000, 'GOMEZ', 'VENDEDOR', 7, 20020102, 1200.00, 10.00, 30), "
        + "(8000, 'BORRERO', 'VENDEDOR', 8, 20020102, 1200.00, 10.00, 30)";

    public static void main(String[] args) {
        /* objeto de conexión */
        Connection conexion = null;
        /* comando para enviar sentencias a través de la conexión */
        Statement sentencia = null;
        try {
            /* carga el driver previamente añadido a las 'Bibliotecas' */
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            /* abre la conexión */
            conexion = DriverManager.getConnection(url, user, password);
            /* crea el comando para enviar sentencias SQL */
            sentencia = conexion.createStatement();
            /* ejecuta la sentencia de creación de la tabla 'empleados' */
            sentencia.executeUpdate(creaEmpleados);
            /* mensaje de confirmación */
            System.out.println("Tabla 'empleados' creada correctamente");
            /* ejecuta la sentencia de inserción de empleados */
            sentencia.executeUpdate(insertaEmpleados);
            /* mensaje de confirmación */
            System.out.println("Nuevos empleados insertados correctamente");
        } catch (Exception ex) {
            /* mensaje de error */

```

```

    System.err.println(ex);
} finally {
    /* si llegó a crearse el comando */
    if (sentencia != null) {
        try {
            /* trata de liberar recursos */
            sentencia.close();
        } catch (SQLException ex) {
        }
    }
    /* si llegó a establecerse la conexión */
    if (conexion != null) {
        try {
            /* trata de liberar recursos */
            conexion.close();
        } catch (SQLException ex) {
        }
    }
}
}
}
}

```

## 4.1 Ejemplo de consulta

Recuerda que para operar con una base de datos, ejecutando las consultas necesarias, nuestra aplicación deberá hacer:

- **Cargar** el driver necesario para comprender el protocolo que usa la base de datos en cuestión.
- **Establecer** una conexión con la base de datos.
- **Enviar** consultas SQL y procesar el resultado.
- **Liberar** los recursos al terminar.
- **Gestionar** los errores que se puedan producir.

Podemos utilizar los siguientes tipos de sentencias:

- **Statement**: para sentencias sencillas en SQL.
- **PreparedStatement**: para consultas preparadas, como por ejemplo las que tienen parámetros.
- **CallableStatement**: para ejecutar procedimientos almacenados en la base de datos.

El API JDBC distingue dos tipos de consultas:

- Consultas: **SELECT**
- Actualizaciones: **INSERT, UPDATE, DELETE**, sentencias DDL

### EJEMPLO 3. Consulta de la tabla departamentos [CreaEmpleados.java]

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
/* clase que ilustra la consulta de la tabla 'departamentos' en la base de datos
 * 'empresaz' del servidor 'localhost', su recogida en un ResultSet, y su
 * posterior impresión en la Salida. */
public class ConsultaDepartamentos {
    /* parámetros de conexión con la base de datos MySQL */
    private static final String url = "jdbc:mysql://localhost/empresaz";
    private static final String user = "root";
    private static final String password = "admin";
    /* sentencia de consulta de la tabla 'departamentos' */
    private static final String consultaDepartamentos =
        "SELECT * FROM departamentos";

    public static void main(String[] args) {
        /* objeto de conexión */
        Connection conexion = null;
        /* comando para enviar sentencias a través de la conexión */
        Statement sentencia = null;
    }
}

```