

# Acceso a una BD MySQL

## Descripción del escenario

La tarea inicial es describir el trabajo que tenemos que abordar. Nos vamos a mover en el contexto de una Universidad. La Universidad oferta titulaciones y se nutre de alumnos. Estos se matriculan en una u otra titulación y cursan un número determinado de asignaturas hasta lograr un número específico de créditos que les proporcionará el título académico correspondiente. Para hacer un seguimiento de los alumnos matriculados, necesitaremos registrar una serie de datos, tanto de ellos como de las asignaturas cursadas.

## Creación de la base de datos

La creación de la base de datos es una tarea específica del sistema administrador de base de datos; en nuestro caso MySQL. El código SQL que nos permite crear esta base de datos es la siguiente:

```
CREATE DATABASE bd_alumnos;
```

Una vez creada la base de datos, el siguiente paso es crear las tablas. Esta operación la vamos a realizar también enviando al sistema administrador de bases de datos las sentencias SQL correspondientes. Previamente, tenemos que informar al sistema administrador de la base de datos que vamos a utilizar:

```
USE bd_alumnos;
```

## Creación de las tablas

La base de datos *bd\_alumnos* está formada por las tres tablas siguientes:

alumnos	
CP	<b>id_alumno</b>
	apellidos nombre curso titulacion

asignaturas	
CP	<b>id_asignatura</b>
	tipo nombre creditos

alumnos_asignaturas	
	id_alumno id_asignatura cursada

Para crear la tabla *alumnos* y asignarle datos, ejecutamos las sentencias SQL siguientes:

```
CREATE TABLE alumnos(  
    id_alumno    INTEGER PRIMARY KEY,  
    apellidos    VARCHAR(24) NOT NULL,  
    nombre       VARCHAR(18) NOT NULL,  
    curso        INTEGER NOT NULL,  
    titulacion   INTEGER NOT NULL,  
);  
  
LOAD DATA LOCAL INFILE "alumnos.txt" INTO TABLE alumnos;
```

alumnos.txt:

9119705	JIMENEZ ALONSO	DIEGO	4	3		
4338289	MANGAS SANZ	CESAR	1	12		
5345629	BARRIGA ASENJO	JOSE	2	7		
5198695	RODRIGUEZ ROBLEDO	FCO. JAVIER			3	5
5434159	BLAZQUEZ BLANCO	SONIA	2	7		

Para crear la tabla *asignaturas* y asignarle datos, ejecutamos las sentencias SQL siguientes:

```
CREATE TABLE asignaturas(  
    id_asignatura    INTEGER PRIMARY KEY,  
    tipo             VARCHAR(2) NOT NULL,  
    nombre           VARCHAR(60) NOT NULL,  
    creditos         FLOAT NOT NULL,  
);  
  
LOAD DATA LOCAL INFILE "asignaturas.txt" INTO TABLE asignaturas;
```

asignaturas.txt:

31540	OB	AMPLIACION DE SISTEMAS OPERATIVOS	4.5	
32330	OP	APLICACIONES DISTRIBUIDAS PARA BIOINGENIERIA		3
33033	OP	APLICACIONES TELEMATICAS	4.5	
20598	TR	ARQUITECTURA DE COMPUTADORES	7.5	
78200	OP	ARQUITECTURA E INGENIERIA DE COMPUTADORES	6	

Y para crear la tabla *alumnos\_asignaturas* y asignarle datos, ejecutamos las sentencias SQL siguientes:

```
CREATE TABLE alumnos_asignaturas(  
    id_alumno        INTEGER NOT NULL,  
    id_asignatura     INTEGER NOT NULL,  
    cursada          CHAR(1) NOT NULL,  
);
```

```
LOAD DATA LOCAL INFILE "alumnos_asignaturas.txt" INTO TABLE
alumnos_asignaturas;
```

alumnos\_asignaturas.txt:

```
9119705      82300  N
9119705      32684  S
4338289      78009  S
4338289      78009  S
5345629      32306  S
5345629      31544  N
```

Para mostrar las bases de datos existentes ejecutamos la orden:

```
SHOW DATABASES;
```

Para verificar la estructura de una tabla ejecutamos la orden:

```
DESCRIBE nombre_tabla;
```

También, podemos mostrar las tablas de la base de datos en uso ejecutando la orden:

```
SHOW TABLES;
```

En lugar de usar un archivo para introducir datos, lo podemos hacer directamente con la orden INSERT INTO. Ejemplo:

```
INSERT INTO alumnos VALUES (324555, 'AGUIRRE SORIANO', 'LETICIA', 4, 3);
```

Para modificar datos en la base de datos, podemos usar la orden UPDATE. Ejemplo:

```
UPDATE alumnos SET nombre = 'JOAQUIN' WHERE id_alumno=9075710;
```

Para borrar un registro de una tabla, usamos la orden DELETE. Ejemplo:

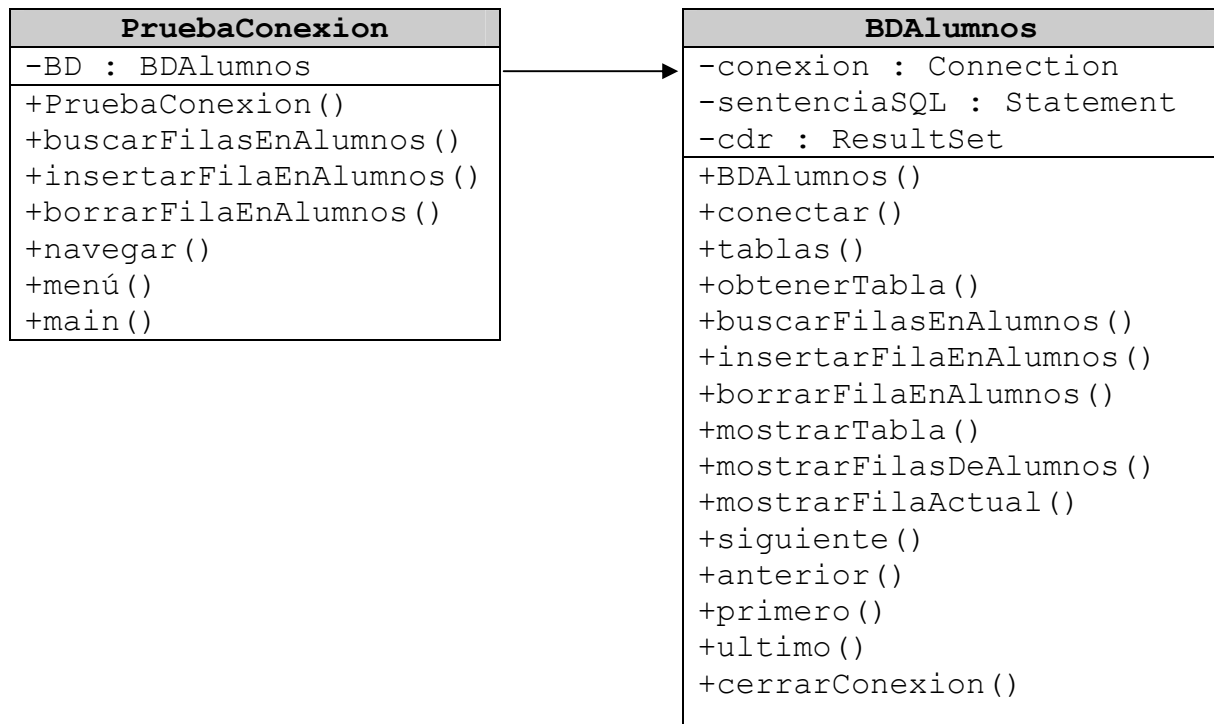
```
DELETE FROM alumnos WHERE id_alumno=324555;
```

Para obtener datos de la base de datos, usamos la orden SELECT. Ejemplo:

```
SELECT * FROM alumnos WHERE curso=1 AND titulacion=12 ORDER BY apellidos;
```

Una vez creada la base de datos y sus tablas, estamos listos para crear una o más clases Java que presenten una interfaz que permita a los usuarios de la base conectarse a la misma para insertar datos o recuperar datos selectivamente, todo ello sin requerirles conocimientos de SQL.

La figura siguiente muestra la estructura de la aplicación que pretendemos crear; está formada por dos clases: **BDAlumnos** y **PruebaConexion**.



La clase **BDAlumnos** encapsulará la base de datos *bd\_alumnos* y mostrará una interfaz pública que permita recuperar datos, insertarlos, borrarlos, etc., y la clase **PruebaConexion**, utilizando la interfaz de **BDAlumnos**, permitirá que cualquier usuario sin conocimientos de SQL ni de bases de datos pueda trabajar con dicha base.

El código de la clase **BDAlumnos** es el siguiente:

```

// Clase base de datos alumnos (BDAlumnos). Métodos:
// constructor BDAlumnos: carga el controlador JDBC y conectar.
// conectar: realiza la conexión con la base de datos.
// tablas: devuelve el conjunto de tablas disponibles en la BD.
// obtenerTabla: obtiene todas las filas de la tabla.
// buscarFilasEnAlumnos: obtiene determinadas filas de la tabla alumnos.
// insertarFilaEnAlumnos: inserta una fila en la tabla alumnos.
// borrarFilaEnAlumnos: borra una fila de la tabla alumnos.
// mostrarTabla: visualiza todas las filas de una tabla.
// mostrarFilasDeAlumnos: visualiza determinadas filas de la tabla alumnos.
// mostrarFilaActual: muestra la fila actual de un ResultSet.
// siguiente: mover el cursor a la fila siguiente.
// anterior: mover el cursor a la fila anterior.
// primero: mover el cursor a la primera fila.
// último: mover el cursor a la última fila.
// cerrarConexion: cerrar la conexión con la base de datos.
  
```

```

public class BDAlumnos
{
    private java.sql.Connection conexión;
    private java.sql.Statement sentenciaSQL;
    private java.sql.ResultSet cdr; // conjunto de resultados

    public BDAlumnos() throws ClassNotFoundException, java.sql.SQLException,
        InstantiationException, IllegalAccessException
    {
        // Cargar el controlador JDBC
        String controlador = "com.mysql.jdbc.Driver";
        Class.forName(controlador).newInstance();
        conectar(); // conectar con la fuente de datos
    }

    public void conectar() throws java.sql.SQLException
    {
        String URL_bd = "jdbc:mysql://127.0.0.1:3306/bd_alumnos";
        String usuario = "";
        String contraseña = "";
        // Conectar con la BD
        conexión = java.sql.DriverManager.getConnection(
            URL_bd, usuario, contraseña);
        // Crear una sentencia SQL
        sentenciaSQL = conexión.createStatement(
            java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE,
            java.sql.ResultSet.CONCUR_UPDATABLE);
        System.out.println("\nConexión realizada con éxito.\n");
        // Mostrar las tablas de la base de datos
        System.out.println("Tablas de la base de datos: ");
        String[] tabla = tablas();
        for (int i = 0; i < tabla.length; ++i)
            System.out.println(tabla[i]);
    }

    public void cerrarConexion() throws java.sql.SQLException
    {
        if (cdr != null) cdr.close();
        if (sentenciaSQL != null) sentenciaSQL.close();
        if (conexión != null) conexión.close();
    }

    public String[] tablas() throws java.sql.SQLException
    {
        cdr = sentenciaSQL.executeQuery("SHOW TABLES");
        cdr.last(); // mover el cursor a la última fila
        String[] tablas = new String[cdr.getRow()];
        cdr.beforeFirst(); // mover el cursor a su posición inicial
        int i = 0;
        while(cdr.next())
            tablas[i++] = cdr.getString(1);
        return tablas;
    }
}

```

```

public java.sql.ResultSet obtenerTabla(String tabla)
    throws java.sql.SQLException
{
    cdr = sentenciaSQL.executeQuery(
        "SELECT * FROM " + tabla);
    return cdr;
}

public java.sql.ResultSet buscarFilasEnAlumnos(String subcad,
    int tipoBúsqueda) throws java.sql.SQLException
{
    String[] cadena = {"'" + subcad + "%'", "'%" + subcad + "%'"};
    cdr = sentenciaSQL.executeQuery(
        "SELECT * FROM " + "alumnos" +
        " WHERE apellidos LIKE " + cadena[tipoBúsqueda-1]);
    return cdr;
}

public void insertarFilaEnAlumnos(int id_alumno,
    String apellidos, String nombre, int curso, int titulación)
    throws java.sql.SQLException
{
    sentenciaSQL.executeUpdate("INSERT INTO " + "alumnos" +
        " VALUES (" + id_alumno + ", '" + apellidos + "', '" +
        nombre + "', " + curso + ", " + titulación + ")"
    );
}

public void borrarFilaEnAlumnos(int ID)
    throws java.sql.SQLException
{
    sentenciaSQL.executeUpdate("DELETE FROM " + "alumnos" +
        " WHERE id_alumno = " + ID);
}

public void mostrarTabla(String tabla)
    throws java.sql.SQLException
{
    cdr = obtenerTabla(tabla);
    while(cdr.next()) mostrarFilaActual();
}

public void mostrarFilasDeAlumnos(String subcad, int tipoBusqueda)
    throws java.sql.SQLException
{
    cdr = buscarFilasEnAlumnos(subcad, tipoBusqueda);
    while(cdr.next()) mostrarFilaActual();
}

public void mostrarFilaActual() throws java.sql.SQLException
{
    int nColumnas = cdr.getMetaData().getColumnCount();
    for (int i = 1; i <= nColumnas; ++i)
    {

```

```

        System.out.print(cdr.getString(i) + " ");
    }
    System.out.println();
}

public void siguiente() throws java.sql.SQLException
{
    if (!cdr.isLast()) cdr.next();
}

public void anterior() throws java.sql.SQLException
{
    if (cdr.isBeforeFirst()) cdr.first();
    if (!cdr.isFirst()) cdr.previous();
}

public void primero() throws java.sql.SQLException
{
    cdr.first();
}

public void último() throws java.sql.SQLException
{
    cdr.last();
}
}

```

**El código de la clase PruebaConexion es el siguiente:**

// Aplicación para probar la base de datos identificada por BD

```

public class PruebaConexion
{
    private static BDAlumnos BD;

    public PruebaConexion() throws ClassNotFoundException,
        java.sql.SQLException, InstantiationException,
        IllegalAccessException
    {
        // Realizar la conexión con la base de datos BD
        BD = new BDAlumnos();
    }

    public void buscarFilasEnAlumnos()
        throws java.sql.SQLException
    {
        System.out.println("\nBuscar:");
        String[] búsquedas = {"cadenas que empiecen por...",
                               "cadenas que contengan..." };
        int i = menú(búsquedas, búsquedas.length);

        System.out.print("> ");
        String subcadena = Leer.dato();
    }
}

```

```

        BD.mostrarFilasDeAlumnos(subcadena, i);
    }

    public void insertarFilaEnAlumnos()
        throws java.sql.SQLException
    {
        int ID, curso, titulación;
        String apellidos, nombre;

        System.out.print("\nID: ");
        ID = Leer.datoInt();
        System.out.print("Apellidos: ");
        apellidos = Leer.dato();
        System.out.print("Nombre: ");
        nombre = Leer.dato();
        System.out.print("Curso: ");
        curso = Leer.datoInt();
        System.out.print("Titulación: ");
        titulación = Leer.datoInt();

        BD.insertarFilaEnAlumnos(ID, apellidos, nombre, curso, titulación);
    }

    public void borrarFilaEnAlumnos()
        throws java.sql.SQLException
    {
        int ID;

        System.out.print("\nIdentificador: ");
        ID = Leer.datoInt();
        BD.borrarFilaEnAlumnos(ID);
    }

    public void navegar() throws java.sql.SQLException
    {
        // Opciones del menú
        String[] opciones = { "Siguiente",
                              "Anterior",
                              "Primero",
                              "Último",
                              "Salir." };

        int opción = 0;
        do
        {
            switch(opción = menú(opciones, opciones.length))
            {
                case 1:
                    BD.siguiente();
                    break;
                case 2:
                    BD.anterior();
                    break;
                case 3:
                    BD.primero();
            }
        }
    }

```



```

        break;
    case 4:
        BD.último();
        break;
    }
    if (opción != 5) BD.mostrarFilaActual();
}
while(opción != 5);
}

public static int menú(String[] opciones, int numOpciones)
{
    int i = 0, opción = 0;

    System.out.println("\n_____ \n");
    for (i = 1; i <= numOpciones; ++i)
    {
        System.out.print("    " + i + ". " + opciones[i-1] + "\n");
    }
    System.out.println("_____ \n");
    do
    {
        System.out.print("\nOpción (1 - " + numOpciones + "): ");
        opción = Leer.datoInt();
    }
    while(opción < 1 || opción > numOpciones);
    return opción;
}

public static void main(String args[])
{
    int i = 0, opción = 0;
    PruebaConexion objAp = null;

    try
    {
        objAp = new PruebaConexion();

        // Opciones del menú
        String[] opciones = { "Datos de la tabla",
                               "Buscar filas en \"alumnos\"",
                               "Insertar fila en \"alumnos\"",
                               "Borrar fila en \"alumnos\"",
                               "Navegar",
                               "Salir." };
        // Nombre de las tablas de la base
        String[] tablas = BD.tablas();

        do
        {
            switch(opción = objAp.menú(opciones, opciones.length))
            {
                case 1:
                    i = objAp.menú(tablas, tablas.length);

```

```

        BD.mostrarTabla(tablas[i-1]);
        break;
    case 2:
        objAp.buscarFilasEnAlumnos();
        break;
    case 3:
        objAp.insertarFilaEnAlumnos();
        break;
    case 4:
        objAp.borrarFilaEnAlumnos();
        break;
    case 5:
        i = objAp.menú(tablas, tablas.length);
        BD.obtenerTabla(tablas[i-1]);
        objAp.navegar();
        break;
    }
}
while (opción != 6);
}
catch(ClassNotFoundException e)
{
    System.out.println(e.getMessage());
}
catch(InstantiationException e)
{
    System.out.print(e.getMessage());
}
catch(IllegalAccessException e)
{
    System.out.print(e.getMessage());
}
catch(java.sql.SQLException e)
{
    System.out.print(e.getMessage());
}
finally // pase lo que pase cerramos la conexión
{
    try
    {
        BD.cerrarConexion();
    }
    catch(java.sql.SQLException ignorada) {}
}
}
}

```