

## Ejercicio 1

En este ejercicio ejecutaremos un primer ejemplo de acceso a bases de datos desde **Java**.

- Utilice la herramienta MySQL Administrator para crear una base de datos **MySQL** llamada Prueba, la cual contenga una única tabla llamada Personas, que tenga una única columna llamada nombre, y que sea clave primaria.
- Ejecute varias veces el primer ejemplo de acceso a bases de datos (disponible en el sitio web del curso), insertando un nombre nuevo en cada ejecución. Luego ejecútelo pretendiendo insertar un nombre repetido y observe los resultados.
- Modifique el ejemplo dado, de modo que ahora driver, url, usuario y password se carguen desde un **archivo de configuración**. Vuelva a ejecutar varias veces el ejemplo.

## Ejercicio 2

En este ejercicio crearemos desde **Java** una base de datos en **MySQL** llamada Escuela, cuyo esquema es el siguiente:

```
Personas (cedula INT, nombre VARCHAR(45), apellido VARCHAR(45))
Maestras (cedula INT, grupo VARCHAR(45))
Alumnos (cedula INT, cedulaMaestra INT)
```

- La columna cedula en Maestras y en Alumnos referencia al campo cedula en Personas.
  - La columna cedulaMaestra en Alumnos referencia al campo cedula en Maestras.
- Escriba un programa Main en **Java** que cree la base de datos según el esquema descrito.
  - Utilice la herramienta MySQL Query Browser para inspeccionar la base de datos creada e insertar algunas tuplas en las tablas creadas.

## Ejercicio 3

En este ejercicio desarrollaremos una aplicación de consola interactiva que permita al usuario ejecutar sentencias de **modificación** (INSERT, UPDATE, DELETE) sobre la BD del ejercicio 2.

```
Ingrese comando: INSERT INTO Personas VALUES (1234567, 'Juan', 'Perez')
Cantidad de filas afectadas: 1
Ingrese comando: DELETE FROM Personas WHERE nombre = 'María'
Cantidad de filas afectadas: 3
Ingrese comando: exit
Hasta la próxima
```

La aplicación leerá comandos de la entrada estándar, los cuales podrán ser sentencias SQL o el comando exit. Puede asumir que el usuario ingresará únicamente sentencias de **modificación**.

Cada comando será leído en un String, el cual será utilizado como texto de la sentencia a ejecutar sobre la base de datos. En caso de tratarse de una sentencia SQL, se mostrará al usuario la cantidad de filas afectadas, o bien el mensaje de error (en caso de tratarse de una sentencia incorrecta). El mensaje de error puede obtenerse consultando el texto de la SQLException correspondiente. En caso de tratarse del comando exit, se finalizará la ejecución.

**Observaciones:**

1. Para poder leer texto de la entrada estándar (consola) utilice los siguientes flujos (streams):
 

```
InputStreamReader is = new InputStreamReader (System.in);
BufferedReader br = new BufferedReader (is);
```
  2. Para leer una línea de texto de consola, utilice el método `br.readLine()`.
- a) Escriba un programa `Main` en **Java** que implemente la aplicación solicitada.
  - b) Ejecútela para diversos comandos del tipo `INSERT`, `UPDATE`, `DELETE` (tanto correctos como incorrectos) y compruebe los resultados utilizando la herramienta `MySQL Query Browser`.

**Ejercicio 4**

Se desea escribir un programa `Main` en **Java** que encuentre la cédula, el nombre y el apellido de la maestra que tiene la **mayor** cantidad de alumnos en la BD del ejercicio 2.

- a) Hágalo usando una **única** consulta SQL que resuelva el problema.
- b) Hágalo usando **tres** consultas SQL diferentes y guardando resultados temporales en memoria.
  - La primera consulta obtiene las cédulas de todas las maestras.
  - La segunda consulta obtiene la cantidad de alumnos de una maestra dada. Realice una iteración en memoria y ejecute dicha consulta para cada maestra devuelta por la primera consulta. Vaya almacenando la cédula de la maestra con mayor cantidad de alumnos hasta el momento.
  - La tercera consulta devuelve finalmente el nombre y el apellido correspondientes a la cédula de la maestra con mayor cantidad de alumnos.
- c) Incorpore **manejo de transacciones** al programa `Main` de la parte (b), de modo que las **tres** consultas SQL se ejecuten dentro de una **misma** transacción.

**Ejercicio 5**

Considere el siguiente **stored procedure** (SP) de **MySQL** a ejecutar sobre la BD del ejercicio 2:

```
CREATE PROCEDURE Escuela.BorrarMaestra (ced INT)
BEGIN
    DECLARE cant INT DEFAULT 0;
    SET cant = (SELECT COUNT(cedula) FROM Maestras m WHERE m.cedula = ced);
    IF cant = 1 THEN
        SELECT a.cedula FROM Alumnos a WHERE a.cedulaMaestra = ced;
        SELECT grupo FROM Maestras m WHERE m.cedula = ced;
        SELECT nombre, apellido FROM Personas p WHERE p.cedula = ced;
        DELETE FROM Alumnos WHERE cedulaMaestra = ced;
        DELETE FROM Maestras WHERE cedula = ced;
        DELETE FROM Personas WHERE cedula = ced;
    END IF;
END
```

- Explique brevemente y con sus propias palabras cuál es el comportamiento del SP propuesto.
- Utilice la herramienta `MySQL Administrator` para crear dicho SP sobre la BD del ejercicio 2.
- Escriba un programa `Main` en **Java** que ejecute el SP propuesto. Debe desplegar en la salida estándar los valores devueltos por cada una de las tres consultas `SELECT` que se ejecutan dentro del bloque `IF` del SP.
- Utilice la herramienta `MySQL Query Browser` para comprobar el estado de la base de datos luego de ejecutado el SP.

## Ejercicio 6

Para hacer este ejercicio, consulte la API de **Java** correspondiente a la interface **DatabaseMetaData** del paquete `java.sql` y busque información en Internet acerca de su funcionamiento.

Se desea escribir una aplicación en **Java** que proporcione un listado en consola de la siguiente información estadística de la instalación de **MySQL**:

- Nombres de las bases de datos existentes en el DBMS.
- Nombres de las tablas de una base de datos determinada, incluyendo los nombres de sus columnas, sus tipos de datos y sus tamaños correspondientes.

Lo primero que hará el programa será brindar un listado de todos los nombres de las bases de datos existentes, el cual deberá tener el siguiente formato:

```
Bases de datos existentes:
    information_schema
    escuela
    prueba
    etc.
```

Posteriormente, el programa solicitará al usuario que ingrese el nombre de una de las bases de datos y listará la información correspondiente a sus tablas según el siguiente formato:

```
Ingrese nombre de la BD a consultar: escuela
Tablas de la BD seleccionada:
    Tabla - alumnos
    Columnas - cedula (INT 10) - cedulaMaestra (INT 10)
    Tabla - maestras
    Columnas - cedula (INT 10) - grupo (VARCHAR 45)
    etc.
```

**Se pide:** Escribir un programa `Main` en **Java** que implemente la aplicación solicitada.

**Observación:** Para poder leer texto de la entrada estándar (consola) utilice los mismos flujos de entrada/salida indicados para el ejercicio 3.