Ejecución de procedimientos (I)

- Los procedimientos almacenados en la base de datos consisten en un conjunto de sentencias SQL y del lenguaje procedural que utiliza el SGBD agrupadas bajo un nombre de procedimiento para realizar tareas sobre la base de datos.
- Se definen una vez y pueden utilizarse cuando se necesiten.
- Pueden definirse con parámetros de entrada (IN), salida (OUT), entrada/salida (IN/OUT) o sin parámetros.
- Si devuelven un valor se denominan **funciones**.
- Dependiendo del SGBD, pueden presentar diferencias, por ejemplo en MySql sólo admiten parámetros IN.

Ejecución de procedimientos (II)

- Ejemplo de procedimiento en MySql:

```
DELIMITER //
CREATE PROCEDURE subida_salario (d INT, subida INT)
BEGIN
 UPDATE empleados SET salario = salario + subida WHERE dept_no=d;
 COMMIT;
END;
//
DELIMITER ;
```

- Se invocaría:

```
CALL subida_salario(20, 100);
```

Ejecución de procedimientos (III)

- Ejemplo de función en MySql:

```
DELIMITER //
CREATE FUNCTION factorial(x INT) RETURNS INT(11)
BEGIN
    DECLARE fact INT;
    SET fact = 1;
    WHILE x > 0 do
            SET fact = fact * x;
            SET x = x - 1;
    END WHILE;
RETURN fact;
END; //
```

- Se ejecutaría de esta forma:

```
SELECT factorial(5)
```

68

Ejecución de procedimientos (IV)

- La interfaz **CallableStatement** permite que se pueda llamar desde Java a los procedimientos almacenados. Para crear un objeto se llama al método prepareCall(String) del objeto *Connection*.
- Ejemplo. Declaración de la llamada al procedimiento subida_salario que tiene dos parámetros y se les da valor con el marcador de posición ?.

```
String sql = "{call subida_salario(?,?)}";
CallableStatement llamada=conexion.prepareCall(sql);
```

Ejecución de procedimientos (V)

- Hay cuatro formas de declarar llamadas a los procedimientos y funciones dependiendo de si hay, o no, parámetros y de la devolución de valores.
- {call procedimiento} para un procedimiento almacenado sin parámetros.
- {?=call function} para una función almacenada que devuelve un valor y no recibe parámetros. El valor se recibe a la izquierda del igual y es el primer parámetro.
- {call procedimiento(?, ?, ...)} para un procedimiento almacenado que recibe parámetros.
- {?=call function(?, ?, ...)} para una función almacenada que devuelve un valor (primer parámetro) y recibe varios parámetros.

Ejecución de procedimientos (VI)

Ejemplo de aplicación que sube el sueldo a los empleados en la base de datos ejemplo de MySql, haciendo uso de un procedimiento almacenado.

```
public class EjecProced {
    public static void main(String[] args) {
        try {
Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/ejemplo", "ejemplo",
"ejemplo");
            int dep = 20;
            int incremento = 111;
            String sql = "{call subida_salario(?,?)}";
            System.out.println(sql);
            CallableStatement llamada=conexion.prepareCall(sql);
            llamada.setInt(1, dep);
            llamada.setInt(2, incremento);
            llamada.execute();
            llamada.close();
            conexion.close();
        } catch (ClassNotFoundException | SQLException cn) { cn.printStackTrace();
        }
    }
}
```