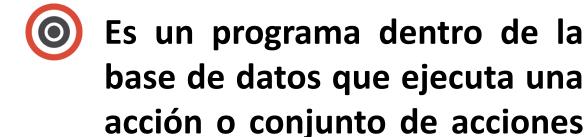
# Conexión Base de Datos

Lenguajes de Programación 2



Dr. Freddy Paz

### **Procedimiento Almacenado**



especificas.



Un procedimiento tiene un nombre, un conjunto de parámetros y un bloque de código.

# **Ejemplo Creación (MySQL)**

```
DELIMITER $$
CREATE PROCEDURE insertarEmpleado(
       IN _dni VARCHAR(8),
       IN nombres VARCHAR(100),
       IN _apellido_paterno VARCHAR(100),
       IN _apellido_materno VARCHAR(100))
BEGIN
       INSERT INTO empleado (dni, nombres, apellido_paterno,
apellido_materno) VALUES (_dni, _nombres, _apellido_paterno,
_apellido_materno);
END
```

### CallableStatement

Es la interfaz utilizada para ejecutar procedimientos SQL almacenados. Los parámetros de entrada se establecen utilizando el método set.

Un objeto de tipo *CallableStatement* puede retornar un objeto de tipo *ResultSet* o un *boolean*. Los parámetros de salida deben registrarse antes de la ejecución del procedimiento almacenado y ser recuperados después de la ejecución.

#### CallableStatement



### import java.sql.CallableStatement;

```
Class.forName("com.mysql.jdbc.Driver");
      Connection con =
DriverManager.getConnection("jdbc:mysql://50.62.209.73/prueba", "prueba",
"lp2");
      CallableStatement cStmt = con.prepareCall("{call
insertarEmpleado(?,?,?,?)}");
      cStmt.setString(1, "29871543");
      cStmt.setString(2, "Freddy Alberto");
      cStmt.setString(3, "Paz");
      cStmt.setString(4, "Espinoza");
      cStmt.execute();
```

#### CallableStatement



```
Class.forName("com.mysql.jdbc.Driver");
      Connection con =
DriverManager.getConnection("jdbc:mysql://50.62.209.73/prueba", "prueba",
"lp2");
      CallableStatement cStmt = con.prepareCall("{call
insertarEmpleado(?,?,?,?)}");
      cStmt.setString(" dni", "12456111");
      cStmt.setString("_nombres", "Freddy Alberto");
      cStmt.setString("_apellido_paterno", "Paz");
      cStmt.setString("_apellido_materno", "Espinoza");
      cStmt.execute();
```

# **Ejemplo Creación Tabla**

```
© CREATE TABLE empleado2 (
id INT AUTO_INCREMENT,
nombres VARCHAR(50),
apellido_paterno VARCHAR(50),
apellido_materno VARCHAR(50),
PRIMARY KEY (id)
);
```

### Ejemplo PA con parámetro salida

```
DELIMITER $$
CREATE PROCEDURE insertarEmpleado2(
       OUT id INT,
       IN nombres VARCHAR(100),
        IN apellido paterno VARCHAR(100),
        IN apellido materno VARCHAR(100))
BEGIN
        INSERT INTO empleado2 (nombres, apellido paterno,
apellido materno) VALUES (nombres, apellido paterno,
_apellido_materno);
       SET id = last insert id();
END
```

### Ejemplo PA con parámetro salida

```
Class.forName("com.mysql.jdbc.Driver");
      Connection con =
DriverManager.getConnection("jdbc:mysql://50.62.209.73/prueba",
"prueba", "lp2");
      CallableStatement cStmt = con.prepareCall("{call
insertarEmpleado2(?,?,?,?)}");
      cStmt.registerOutParameter("_id", java.sql.Types.INTEGER);
      cStmt.setString("_apellido_paterno", "Paz");
      cStmt.setString("_nombres", "Freddy Alberto");
      cStmt.setString("_apellido_materno", "Espinoza");
      cStmt.execute();
      int id = cStmt.getInt("_id");
      System.out.println(id);
```

### Ejemplo PA con parámetro salida



```
MySqlConnection con = new MySqlConnection(cadena);
con.Open();
MySqlCommand comando = new MySqlCommand();
comando.Connection = con;
comando.CommandType = CommandType.StoredProcedure;
comando.CommandText = "INSERTAR PROFESOR";
comando.Parameters.Add("_nombre", MySqlDbType.VarChar).Value = "Juan";
comando.Parameters.Add("_apellido", MySqlDbType.VarChar).Value = "Perez";
comando.Parameters.Add(" id profesor", MySqlDbType.Int32).Direction = ParameterDirection.Output;
comando.ExecuteNonQuery();
int numero = Int32.Parse(comando.Parameters["_id_profesor"].Value.ToString());
System.Console.WriteLine(numero);
System.Console.Read();
con.Close();
```

# **PA listar Empleados**

```
DELIMITER $$
CREATE PROCEDURE listarEmpleados()
BEGIN
SELECT * FROM empleado;
END
```

# Listar Empleados - Código

```
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://50.62.209.73/prueba",
"prueba", "lp2");
CallableStatement cStmt = con.prepareCall("{call
       listarEmpleados()}");
ResultSet rs = cStmt.executeQuery();
while (rs.next()){
        String dni = rs.getString("dni");
        System.out.println(dni);
```

### PA buscar Empleados

```
DELIMITER //
CREATE PROCEDURE buscarEmpleado(
      IN variable VARCHAR(50)
BEGIN
      SELECT * from
                        empleado where nombres
                                                    like
Concat('%',variable,'%');
END
```

## Creación de Tabla (MSSQL)

```
© CREATE TABLE CLIENTE(
ID_CLIENTE INT PRIMARY KEY IDENTITY(1,1),
NOMBRE VARCHAR(100),
APELLIDO VARCHAR(100)
)
```

# **Ejemplo Creación (MSSQL)**

```
CREATE PROCEDURE INSERTAR_CLIENTE
      @ID INT OUTPUT,
      @NOMBRE VARCHAR(100),
      @APELLIDO VARCHAR(100)
AS
      INSERT INTO CLIENTE(NOMBRE, APELLIDO)
      VALUES (@NOMBRE,@APELLIDO)
      SET @ID = @@IDENTITY
GO
```

## **Ejecutar Procedimiento (MSSQL)**

DECLARE @ID INT
EXEC INSERTAR\_CLIENTE @ID OUTPUT, 'Juan','Perez'
SELECT @ID

## **Ejecutar Procedimiento (MSSQL)**

DECLARE @ID INT
EXEC INSERTAR\_CLIENTE @ID OUTPUT, 'Juan','Perez'
SELECT @ID