

CURSO DE JAVA CON JDBC

EJERCICIO

STORED PROCEDURES CON CALLABLE STATEMENT DE JDBC

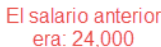


Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

Vamos a crear un stored procedure de Oracle para poner en práctica el tema de Callable Statement con JDBC. Al finalizar veremos:



PASO 1. COPIAMOS EL CÓDIGO DE LA FUNCIÓN DE ORACLE

Archivo procedimientoSetEmployeeSalary.sql:

```
--stored procedure que actualiza el salario de un id_empleado
--segun el factor de incremento proporcionado
CREATE OR REPLACE PROCEDURE "HR"."SET_EMPLOYEE_SALARY" (
    p_emp_id    IN employees.employee_id%TYPE
    , p_factor   IN NUMBER
) AS

    v_monthly_salary  employees.salary%TYPE;

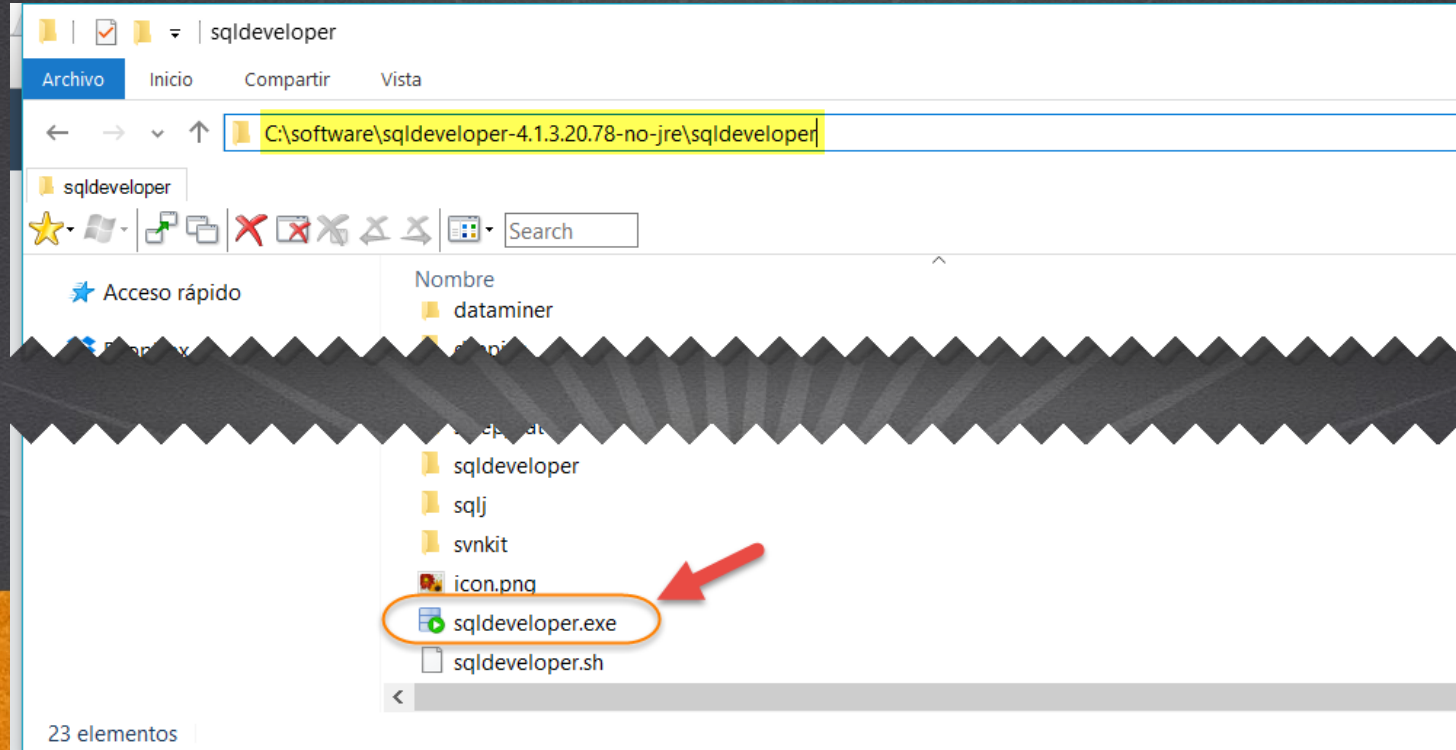
BEGIN

    -- Obtiene el salario actual del id_empleado proporcionado
    SELECT NVL(salary, -999)
        INTO    v_monthly_salary
        FROM    employees
        WHERE   employee_id = p_emp_id;

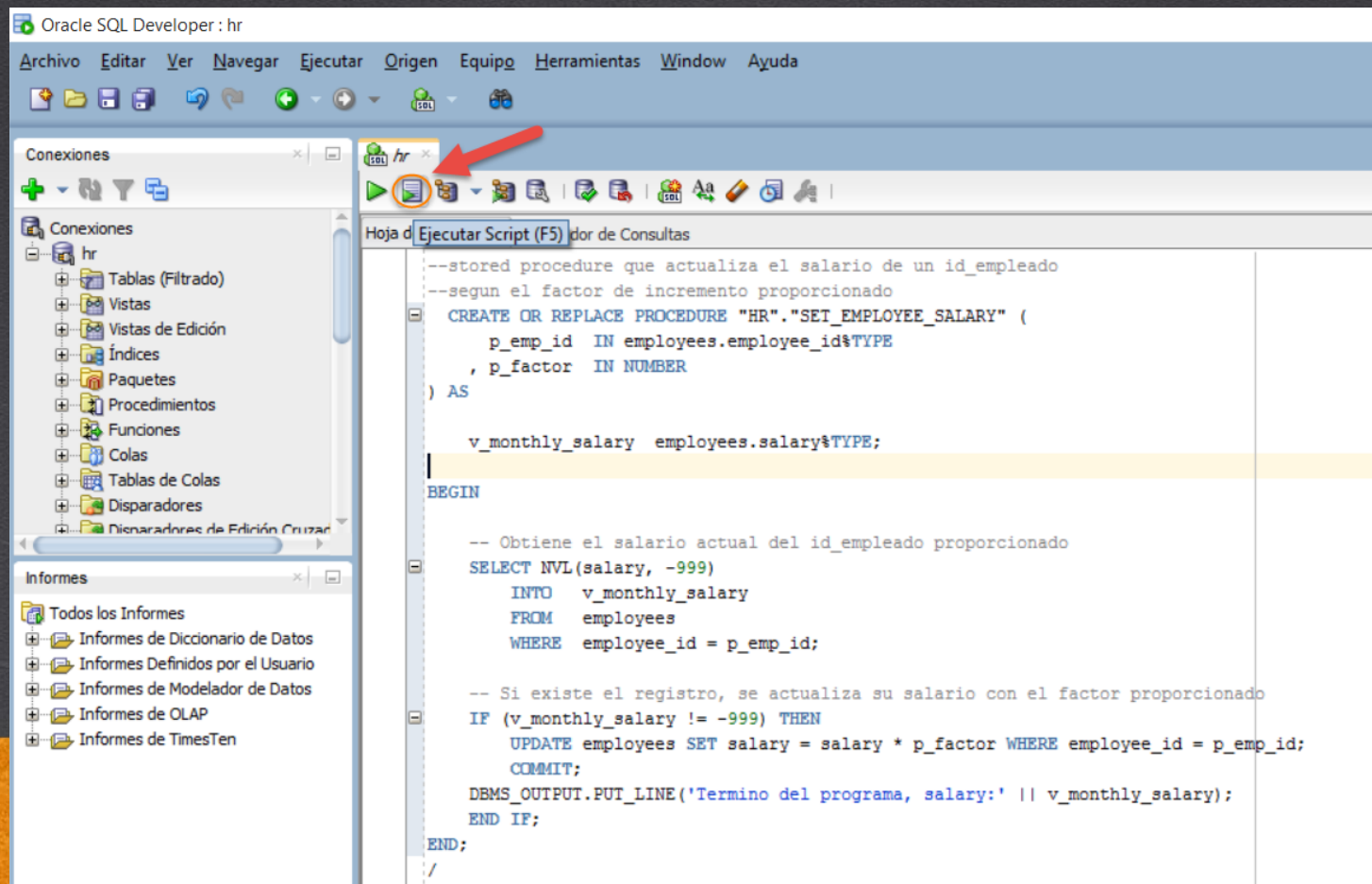
    -- Si existe el registro, se actualiza su salario con el factor proporcionado
    IF (v_monthly_salary != -999) THEN
        UPDATE employees SET salary = salary * p_factor WHERE employee_id = p_emp_id;
        COMMIT;
    DBMS_OUTPUT.PUT_LINE('Termino del programa, salary:' || v_monthly_salary);
    END IF;
END;
/
```


PASO 2. ABRIMOS SQL DEVELOPER

Abrimos el programa de SQL Developer:



PASO 3. PEGAMOS EL PROCEDIMIENTO Y LO EJECUTAMOS



PASO 4. RESULTADO DE LA EJECUCIÓN DEL PROCEDIMIENTO

The screenshot displays the Oracle SQL Developer interface with the 'hr' schema selected. The left pane shows the 'Conexiones' (Connections) tree with 'Procedimientos' (Procedures) expanded, highlighting 'SET_EMPLOYEE_SALARY'. The main editor shows the SQL script for creating or replacing the procedure. The script includes comments in Spanish, parameter definitions, and a call to the procedure. The bottom pane shows the 'Salida de Script' (Script Output) window with the message 'Procedure SET_EMPLOYEE_SALARY compilado' (Procedure SET_EMPLOYEE_SALARY compiled).

Oracle SQL Developer : hr

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Window Ayuda

Conexiones

- Conexiones
 - hr
 - Tablas (Filtrado)
 - Vistas
 - Vistas de Edición
 - Índices
 - Paquetes
 - Procedimientos
 - ADD_JOB_HISTORY
 - SECURE_DML
 - SET_EMPLOYEE_SALARY
 - Funciones
 - Colas

hr

0.044 segundos

Hoja de Trabajo Generador de Consultas

```
--stored procedure que actualiza el salario de un id_empleado
--segun el factor de incremento proporcionado
CREATE OR REPLACE PROCEDURE "HR"."SET_EMPLOYEE_SALARY" (
    p_emp_id  IN employees.employee_id%TYPE
    , p_factor IN NUMBER
) AS

    v_monthly_salary employees.salary%TYPE;

BEGIN

    -- Obtiene el salario actual del id_empleado proporcionado
    SELECT NVL(salary, -999)
```

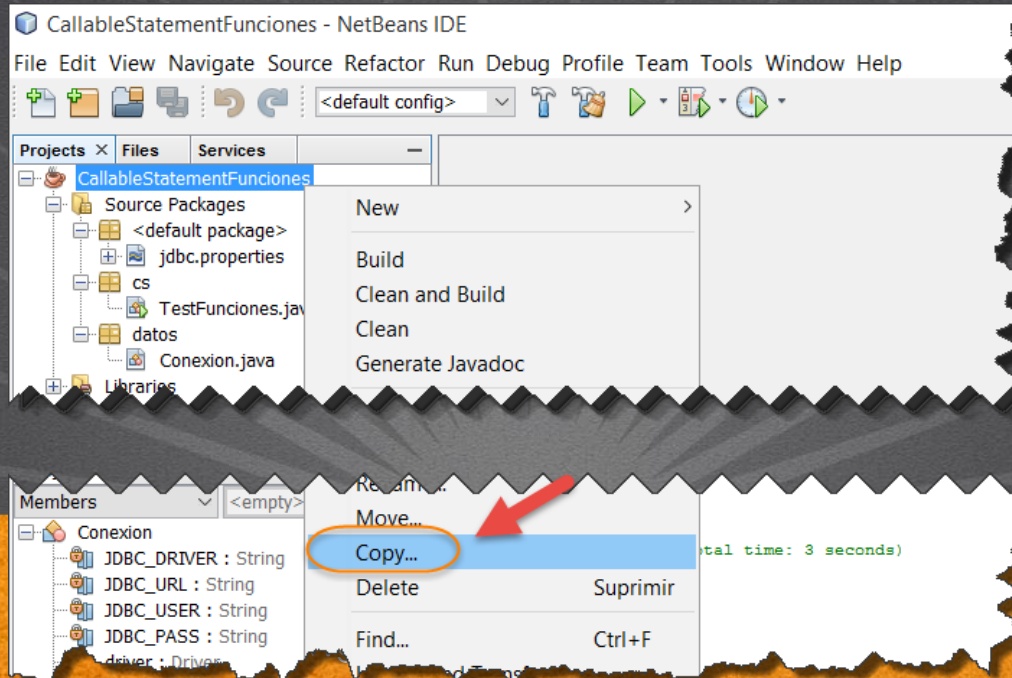
Salida de Script x

Tarea terminada en 0.044 segundos

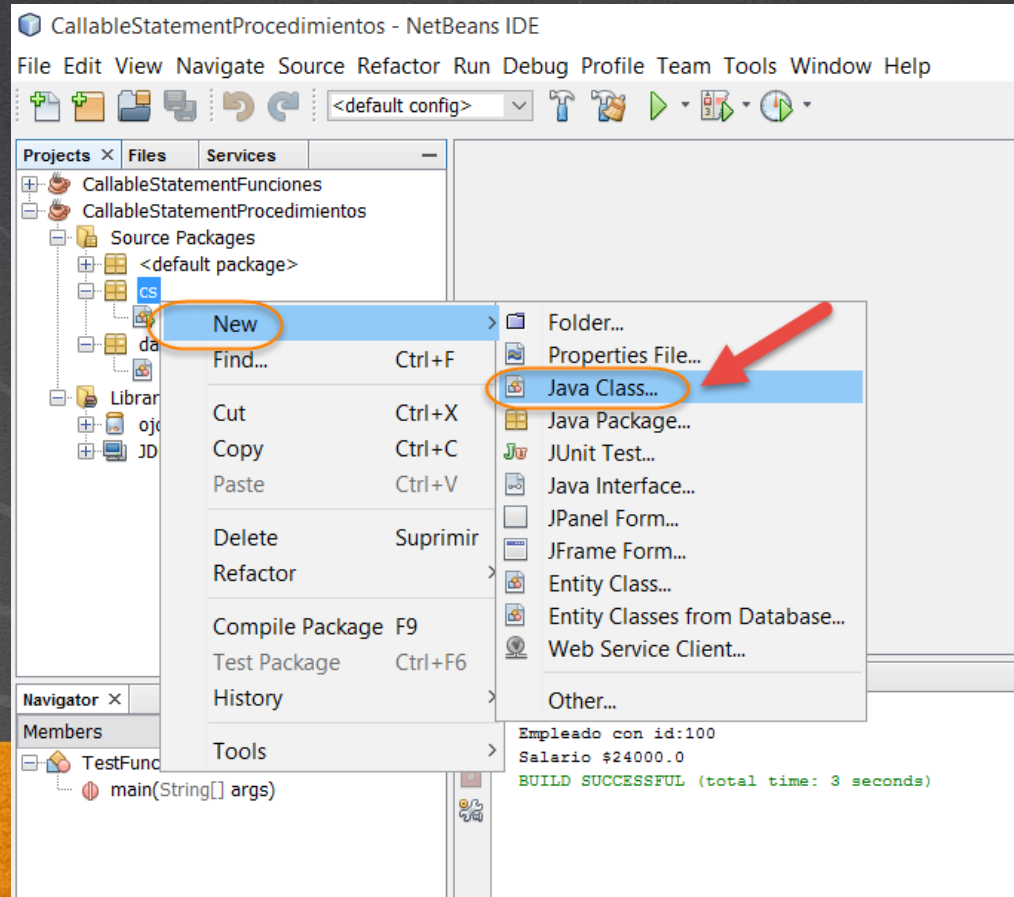
Procedure SET_EMPLOYEE_SALARY compilado

PASO 5. COPIAMOS UN PROYECTO JAVA

Copiamos el proyecto CallableStatementFunciones, si no es posible, solo copiamos las clases de Conexión, el archivo de propiedades jdbc y agregamos el driver de Oracle al classpath como lo hicimos en el proyecto anterior:



PASO 6. CREAMOS UNA CLASE JAVA



PASO 6. CREAMOS UNA CLASE JAVA

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo TestProcedimientos.java:

```
package cs;

import datos.Conexion;
import java.sql.*;

public class TestProcedimientos {

    public static void main(String[] args) {
        int empleadoId = 100;
        //la formula que aplica el Store Procedure es
        //salario=salario*incremento
        double incrementoSalario = 1.1; //incremento del 10%
        Connection con;
        try {
            con = Conexion.getConnection();
            Statement stmt = null;
            ResultSet rset = null;
            CallableStatement cstmt = null;

            stmt = con.createStatement();

            //Llamamos al SP para incrementar el salario
            System.out.println("Aumento del 10% al empleado:" + empleadoId);
            cstmt = con.prepareCall("{call set_employee_salary(?,?)}");
            cstmt.setInt(1, empleadoId);
            cstmt.setDouble(2, incrementoSalario);
            cstmt.execute();
            cstmt.close();
```

```
        //Obtenemos el nuevo valor del salario para el
        empleado seleccionado
        String query = "SELECT first_name, salary FROM
        employees "
                        + " WHERE employee_id = " + empleadoId;

        rset = stmt.executeQuery(query);
        rset.next();
        System.out.println("Nombre: " + rset.getString(1));
        System.out.println("Salario nuevo:" +
        rset.getFloat(2));

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

PASO 8. EJECUTAR EL CÓDIGO

CallableStatementProcedimientos - NetBeans IDE

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

- CallableStatementFunciones
- CallableStatementProcedimientos
 - Source Packages
 - <default package>
 - cs
 - TestFunciones.java
 - TestProcedimientos.java
 - datos
 - Conexion.java

TestProcedimientos.java

Source History

```
22     ResultSet rset = null;
23     CallableStatement cstmt = null;
24
25     stmt = con.createStatement();
26
27     //Llamamos al SP para incrementar el salario
28     System.out.println("Aumento del 10% al empleado:" + empleado);
29
30
31     rset.next();
32     System.out.println("Nombre: " + rset.getString(1));
33     System.out.println("Salario nuevo:" + rset.getFloat(2));
34
35     catch (SQLException e) {
```

cs.TestProcedimientos > main > try >

Output - CallableStatementProcedimientos (run) x

run:

```
Aumento del 10% al empleado:100
Nombre: Steven
Salario nuevo:26400.0
BUILD SUCCESSFUL (total time: 4 seconds)
```

El salario anterior era: 24,000

PASO 9. VERIFICAMOS RESULTADO

Oracle SQL Developer : Tabla HR.EMPLOYEES@hr

Archivo Editar Ver Navegar Ejecutar Equipo Herramientas Window Ayuda

Conexiones

Conexiones

hr

Tablas (Filtrado)

- COUNTRIES
- DEPARTMENTS
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS

EMPLOYEES

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL

Ordenar... Filtrar:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
1	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	26400	
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	17000	
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/06	IT_PROG	9000	
5	104	Bruce	Ernst	BERNST	590.423.4568	21/05/07	IT_PROG	6000	
6	105	David	Austin	DAUSTIN	590.423.4569	25/07/05	IT_PROG	4800	

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio pusimos en práctica el concepto de Callable Statement, y en particular cómo ejecutar un procedimiento almacenado de Oracle.
- La diferencia entre una función y un stored procedure básicamente es que la función regresa un resultado, en cambio el stored procedure no regresa ninguna información. Esto es la base para mandar a llamar cualquier función o procedimiento almacenado en Oracle desde Java.



Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

CURSO ONLINE

JAVA CON JDBC

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx