

Acceso a datos

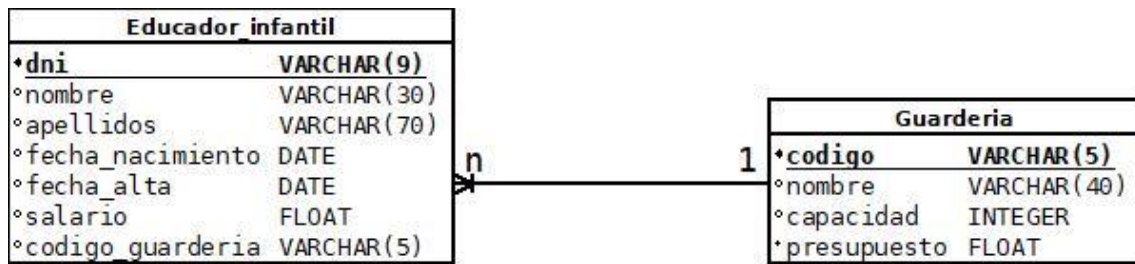
Tarea Unidad 2

Contenido

Introducción.....	2
Ejercicio 1.....	2
Nota aclaratoria.....	5
Ejercicio 2.....	5
Ejercicio 3.....	9

Introducción

La base de datos centrosEducacionInfantil tiene la siguiente estructura:



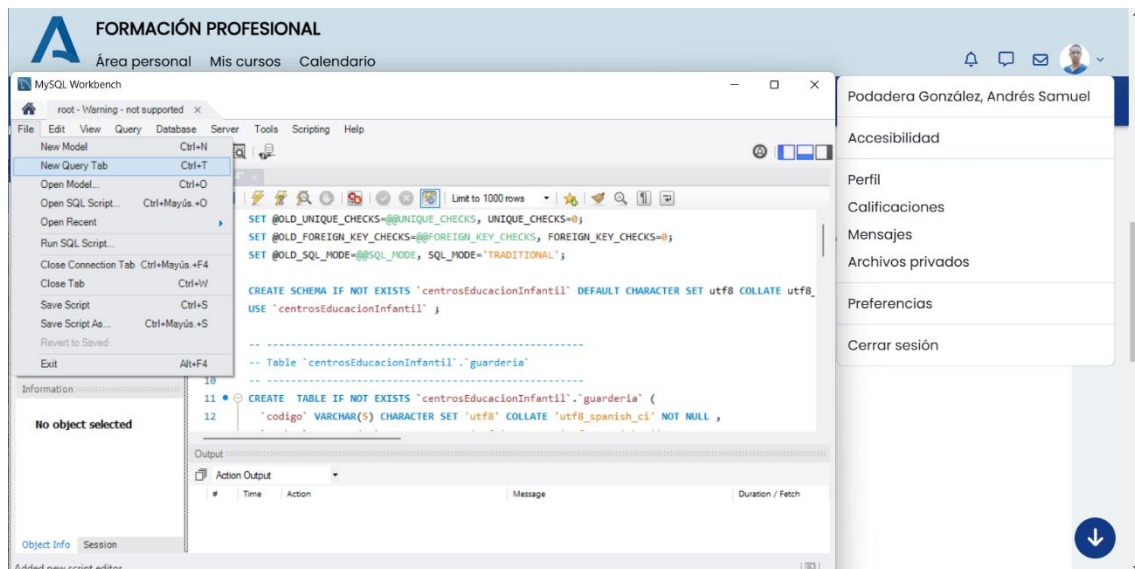
A partir de dicha base de datos, realiza los siguientes ejercicios:

Ejercicio 1

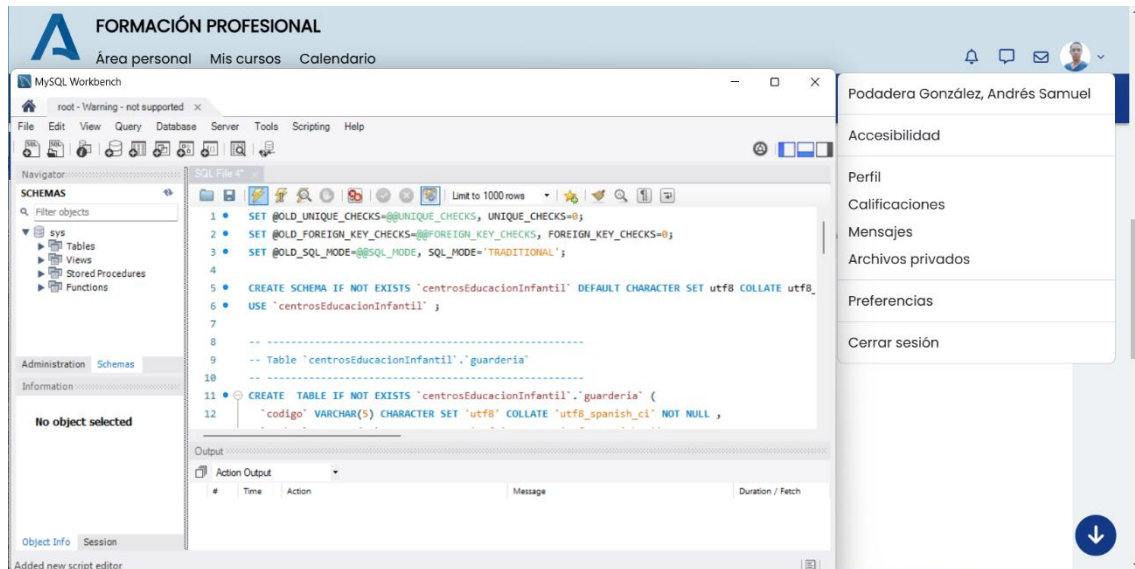
Lanza el script SQL de creación del esquema de la base de datos (Educacion_Infantil) y asegúrate de que se crean ambas tablas y se insertan los registros correspondientes (al menos 10 registros en la tabla educador_infantil y 4 en la tabla guardería). Puedes usar, bien un cliente gráfico (MySQL Workbench) o bien un cliente en modo texto (Línea de comandos MySQL).

Para realizar este ejercicio usaré MySQL Workbench 8.0. Lo primero será arrancar la aplicación y conectarnos a la base de datos con nuestras credenciales.

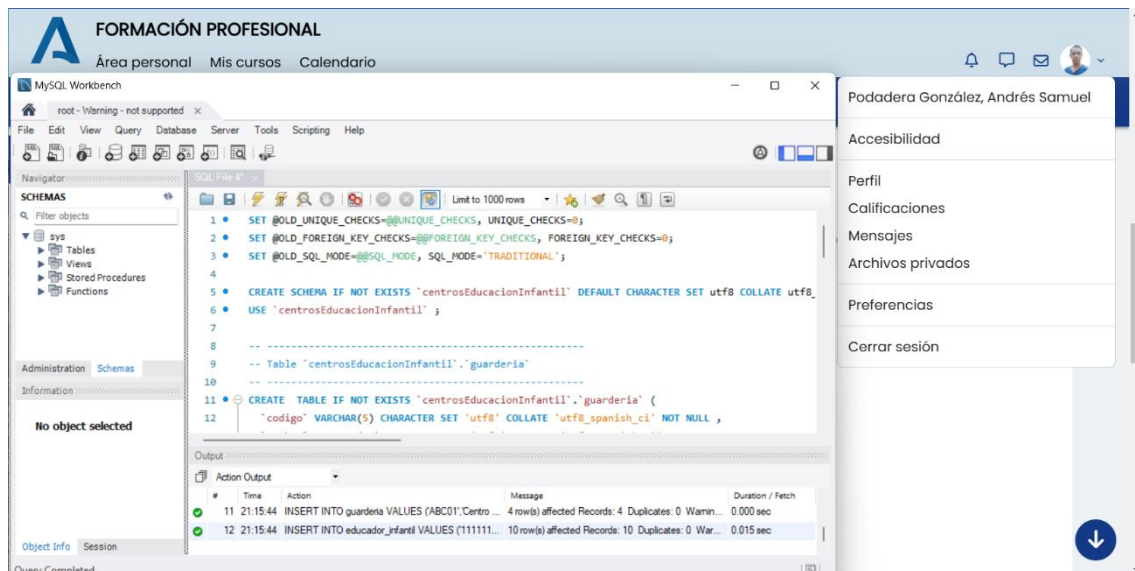
Una vez dentro abrimos una pestaña nueva donde ejecutaremos el script:



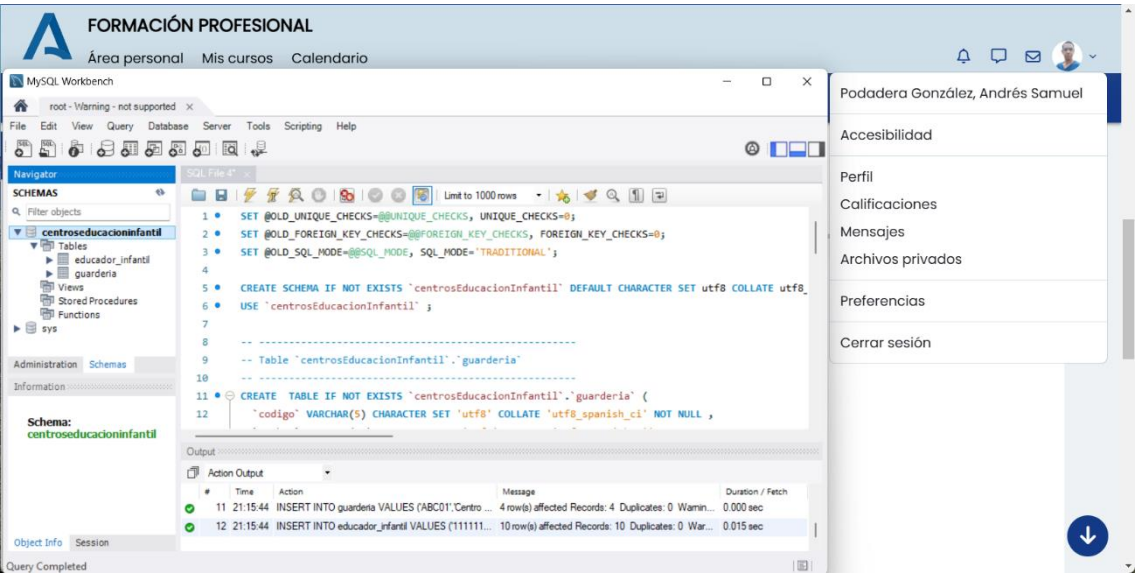
En esa pestaña tendremos que copiar el contenido de nuestro script y hacer clic en el símbolo del “rayo” como se ve en la siguiente imagen para ejecutar el script:



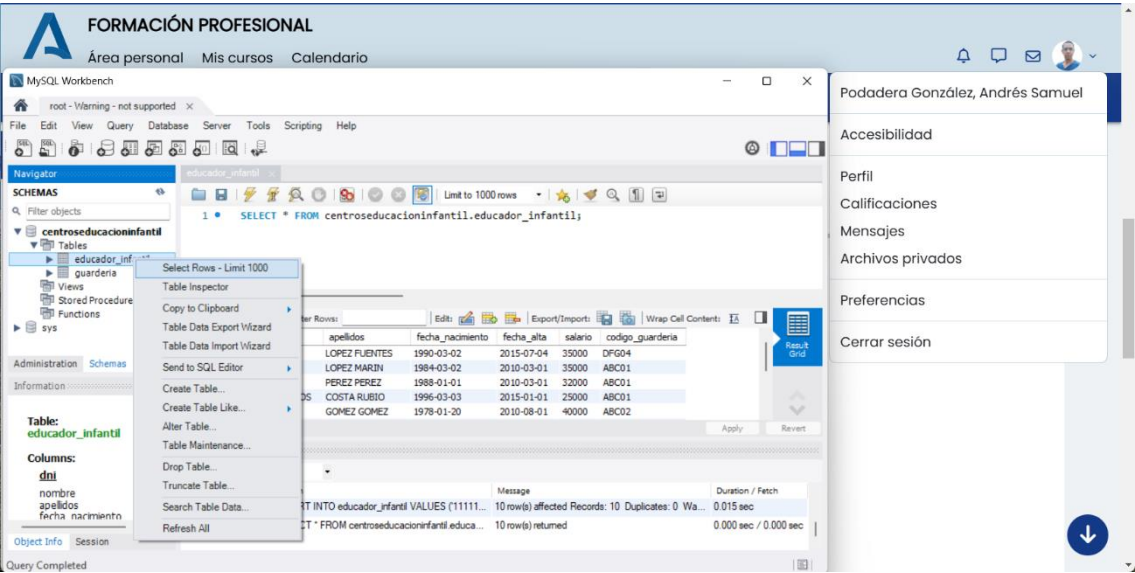
Acabada la ejecución, en la zona inferior aparecerán los mensajes de operación correcta y las filas que se han creado en la base de datos.



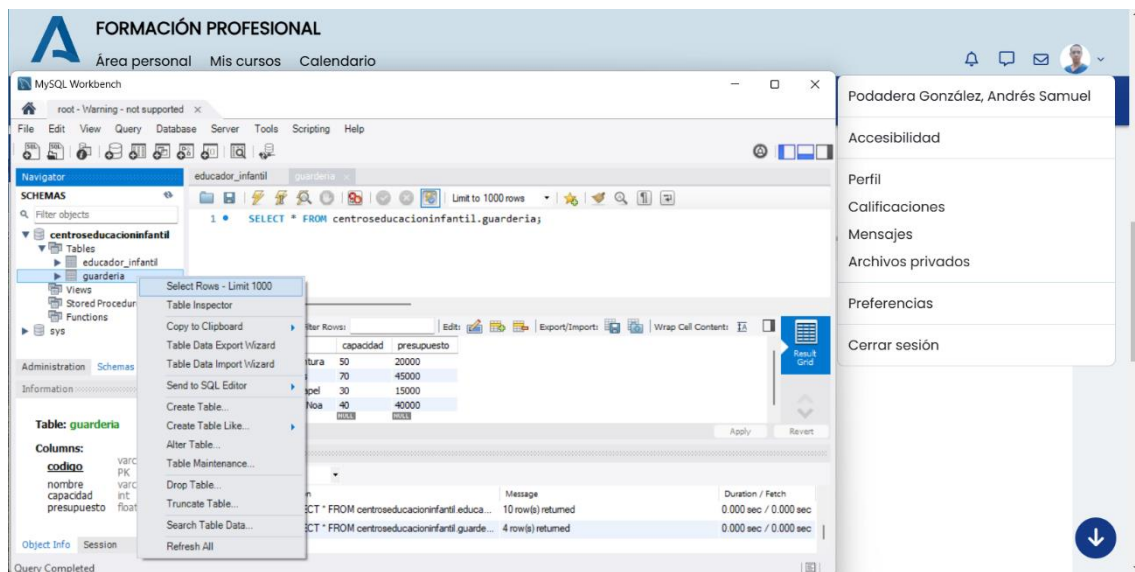
Para asegurarnos de que los datos se han introducido correctamente en la base de datos vamos a proceder a consultarlos. Primero refrescamos la zona lateral izquierda de “Schemas” haciendo clic sobre el icono de flechas girando(al lado de la palabra Schemas) y podremos ver el nombre de la base de datos y sus tablas:



Haciendo clic derecho sobre la primera tabla educador_infantil, seleccionamos la primera opción y listamos todo el contenido de la tabla:



Realizamos la misma operación en la segunda tabla guardería y mostramos todos sus registros:



Como se puede observar el script se ejecuto correctamente y tenemos la información almacenada en la base de datos.

Nota aclaratoria

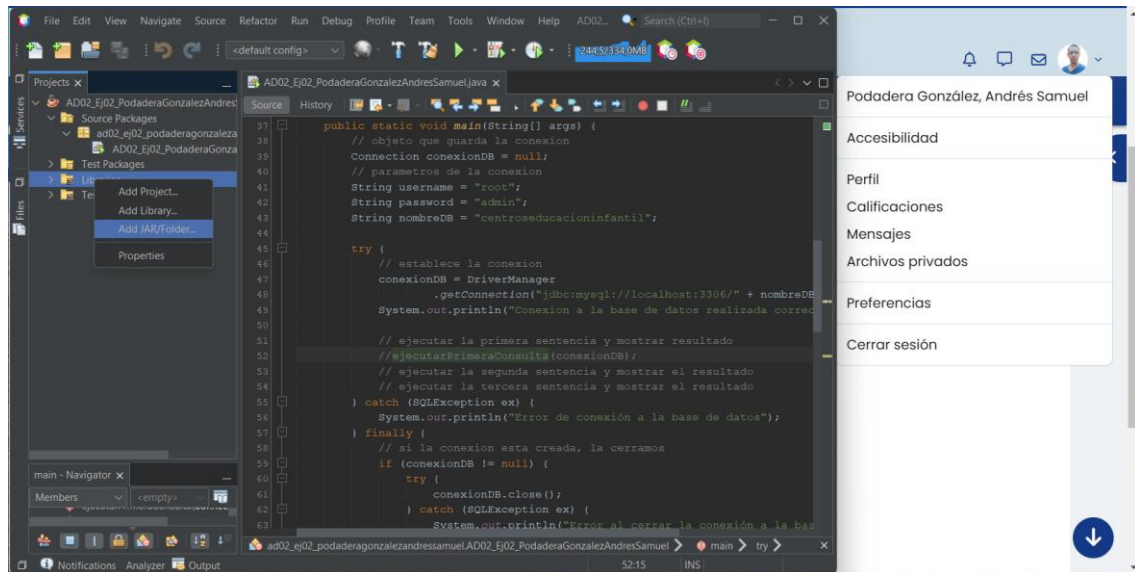
Para los ejercicios prácticos usando la base de datos MySQL he tenido que cambiar el puerto por defecto al 3307 para evitar conflictos con otras bases de datos y contenedores que tengo en mi máquina.

Ejercicio 2

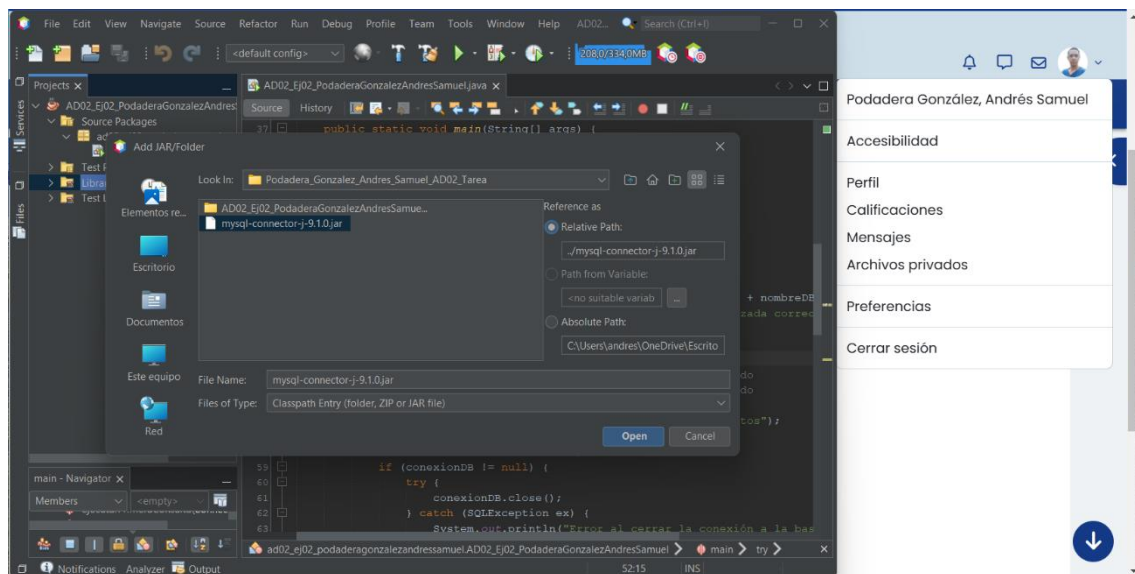
Crea un proyecto en Netbeans con nombre AD02_Ej02_ApellidosNombre que establezca conexión con la BD y muestre la información obtenida, resultado de implementar a través de sentencias preparadas y parametrizadas las siguientes **consultas**, por la salida estándar:

Una vez creado el proyecto vamos a agregar el conector MySQL(está localizado en la raíz del proyecto) para que nuestro programa pueda conectarse a la base de datos. Haciendo clic derecho sobre las librerías del proyecto, seleccionamos la opción “Add Jar/folder...”:

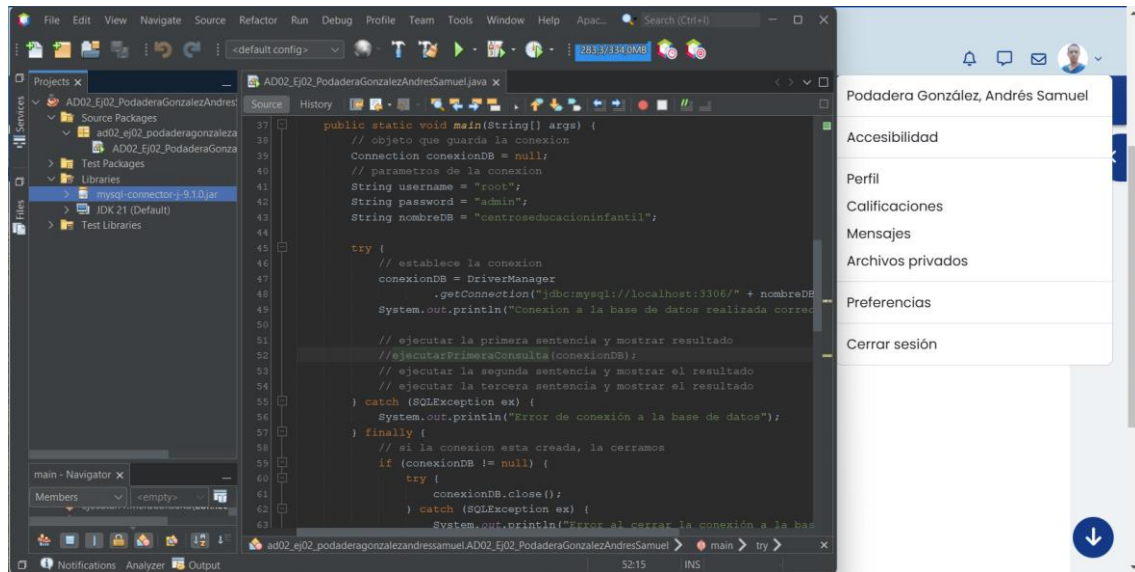
Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025



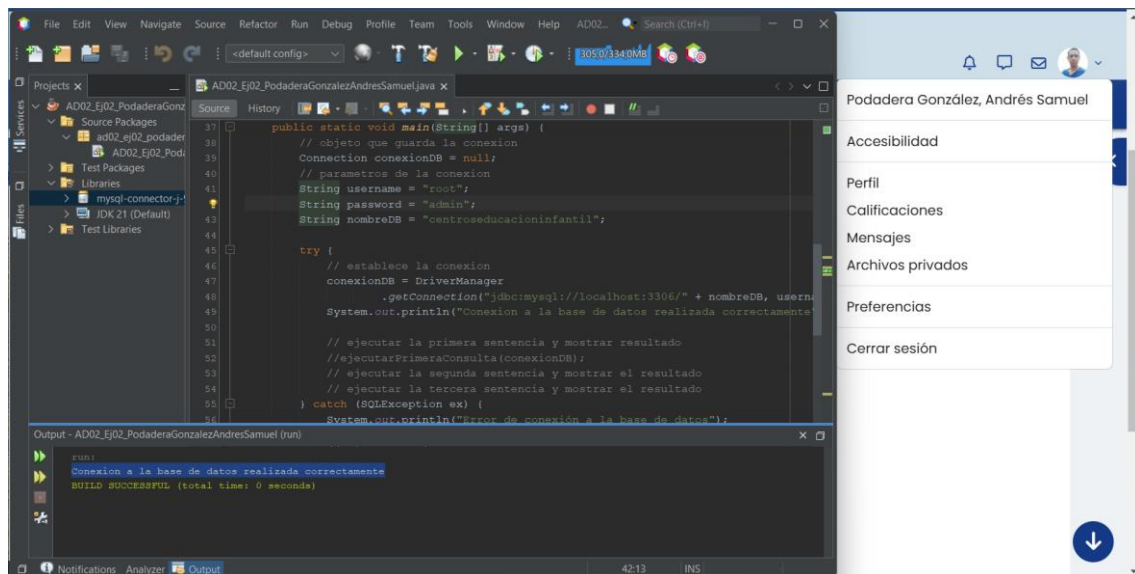
Elegimos el conector específico para nuestra base de datos, descargado desde <https://dev.mysql.com/downloads/connector/j/?os=26> y lo agregamos:



Comprobamos que está añadido a las librerías de nuestro proyecto:

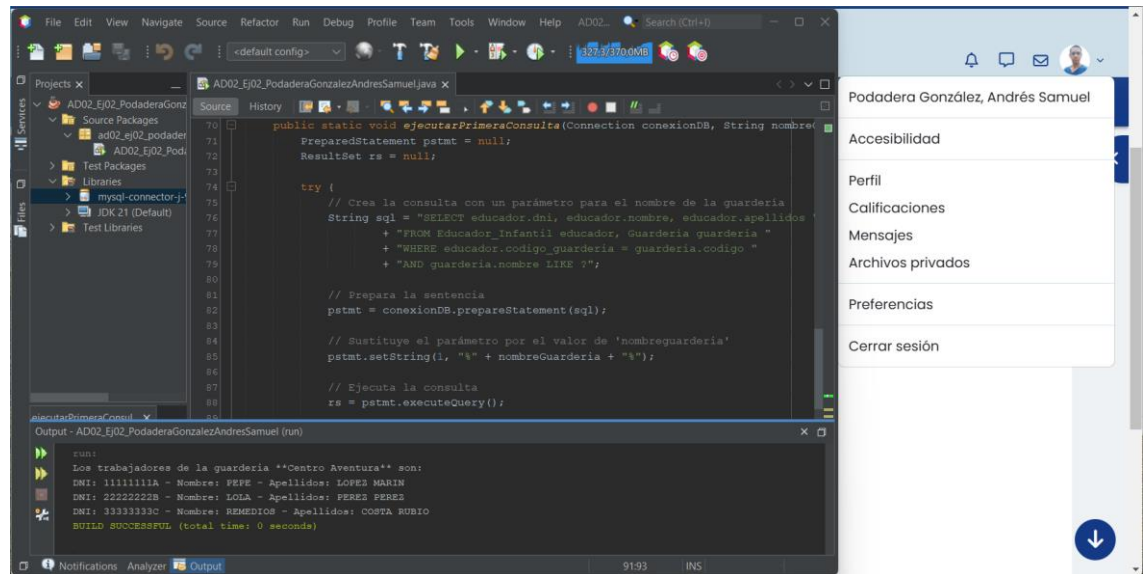


Y ahora vamos a comprobar si nuestro programa se conecta correctamente a la base de datos:



Así es, nos muestra el mensaje de conexión realizada correctamente. Para las siguientes partes de este ejercicio comentaré esa línea y así quedará más limpia la salida de las consultas.

- a. **Consulta 1:** Educadores que trabajan en una guardería indicada por su nombre.



```
public static void ejecutarPrimeraConsulta(Connection conexionDB, String nombre) {
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        // Crea la consulta con un parámetro para el nombre de la guardería
        String sql = "SELECT educador.dni, educador.nombre, educador.apellidos "
            + "FROM Educador_Infantil educador, Guarderia guarderia "
            + "WHERE educador.codigo_guarderia = guarderia.codigo "
            + "AND guarderia.nombre LIKE ?";

        // Prepara la sentencia
        pstmt = conexionDB.prepareStatement(sql);

        // Sustituye el parámetro por el valor de 'nombreguarderia'
        pstmt.setString(1, "%" + nombreGuarderia + "%");

        // Ejecuta la consulta
        rs = pstmt.executeQuery();

        // ... (rest of the code) ...
    } catch (SQLException e) {
        // ... (error handling) ...
    }
}
```

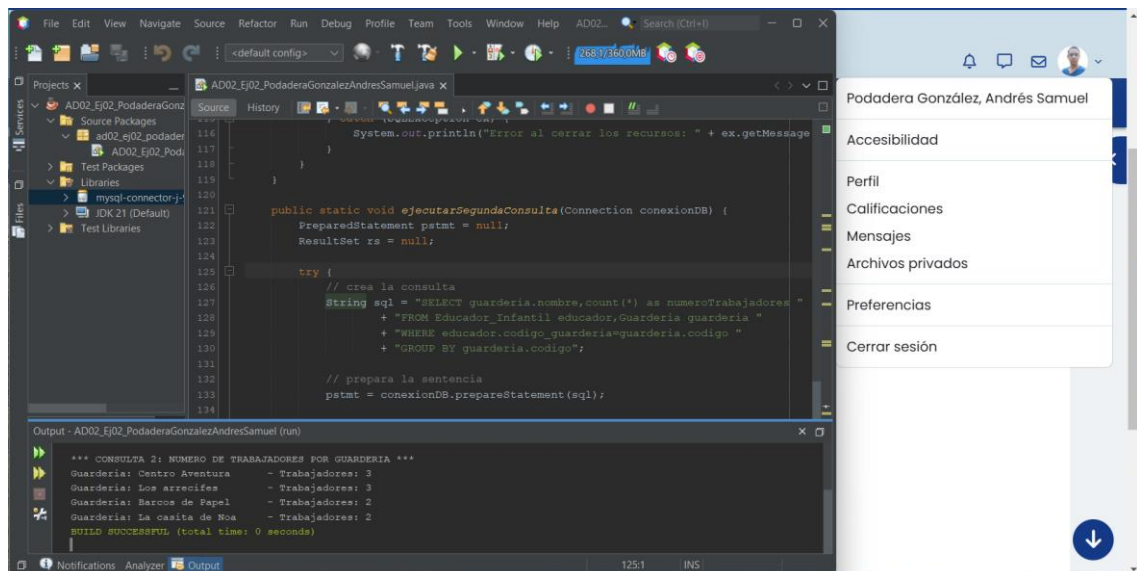
Output - AD02_Ej02_PodaderaGonzalezAndresSamuel (run)

```
run:
Los trabajadores de la guarderia "Centro Aventura" son:
DNI: 11111111A - Nombre: FEVE - Apellidos: LOPES MARIN
DNI: 22222222B - Nombre: LOLA - Apellidos: PEREZ PEREZ
DNI: 33333333C - Nombre: REMEDIOS - Apellidos: COSTA RUBIO
BUILD SUCCESSFUL (total time: 0 seconds)
```

Usamos la sentencia(veo conveniente sólo mostrar dni, nombre y apellidos):

SELECT educador.dni, educador.nombre, educador.apellidos
FROM Educador_Infantil educador, Guarderia guarderia
WHERE educador.codigo_guarderia = guarderia.codigo
AND guarderia.nombre LIKE ?;

- b. **Consulta 2:** Para cada guardería, obtener el número de educadores que trabajan en dicha guardería.



```
public static void ejecutarSegundaConsulta(Connection conexionDB) {
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        // crea la consulta
        String sql = "SELECT guarderia.nombre, count(*) as numeroTrabajadores "
            + "FROM Educador_Infantil educador, Guarderia guarderia "
            + "WHERE educador.codigo_guarderia=guarderia.codigo "
            + "GROUP BY guarderia.codigo";

        // prepara la sentencia
        pstmt = conexionDB.prepareStatement(sql);

        // ... (rest of the code) ...
    } catch (SQLException e) {
        // ... (error handling) ...
    }
}
```

Output - AD02_Ej02_PodaderaGonzalezAndresSamuel (run)

```
*** CONSULTA 2: NUMERO DE TRABAJADORES POR GUARDERIA ***
Guarderia: Centro Aventura - Trabajadores: 3
Guarderia: Los arrecifes - Trabajadores: 3
Guarderia: Barcos de Papel - Trabajadores: 2
Guarderia: La casita de Moa - Trabajadores: 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

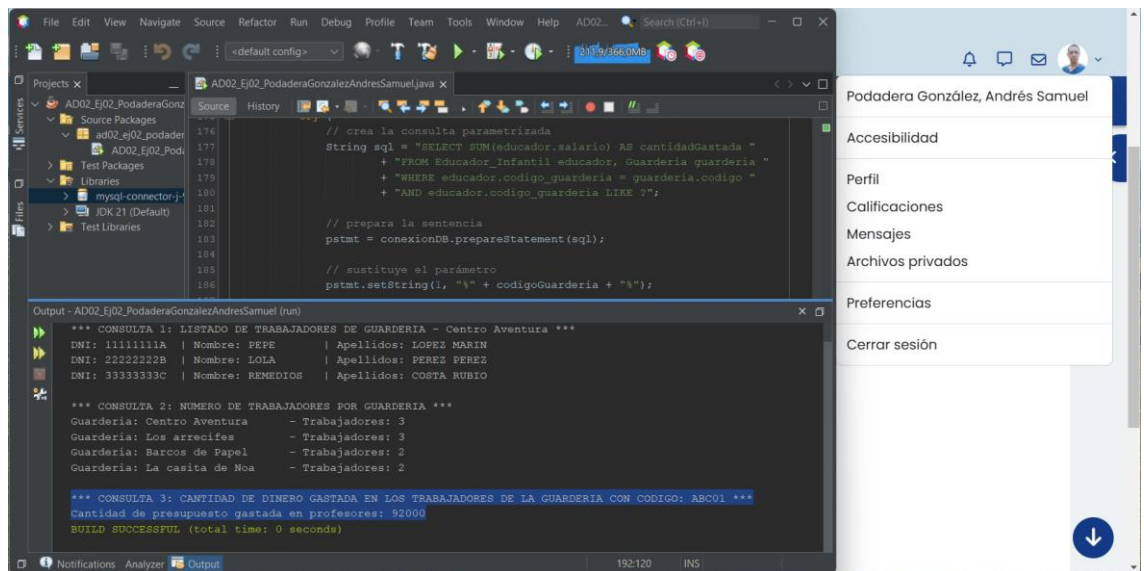
Usamos la sentencia:

SELECT guarderia.nombre,count(*) AS numeroTrabajadores
FROM Educador_Infantil educador,Guarderia guarderia

WHERE educador.codigo_guarderia=guarderia.codigo

GROUP BY guarderia.codigo;

- c. **Consulta 3:** Calcular la cantidad de presupuesto (cantidad de dinero) gastado en el salario de los educadores de una determinada guardería que se pasará como parámetro (se pasará su código).



Usando la sentencia:

SELECT SUM(educador.salario) AS cantidadGastada

FROM Educador_Infantil educador, Guarderia guarderia

WHERE educador.codigo_guarderia = guarderia.codigo

AND educador.codigo_guarderia LIKE ?;

Debes gestionar las posibles excepciones y errores que puedan presentarse, así como el cierre de recursos utilizados de forma adecuada si la aplicación dejara de funcionar (usa la sentencia *try-catch-finally* o equivalente).

Ejercicio 3

Desarrolla un proyecto en NetBeans con nombre `AD02_Ej03_ApellidoNombre` que establezca conexión con la BD y muestre la información obtenida, resultado de ejecutar los siguientes **procedimientos almacenados**, por la salida estándar:

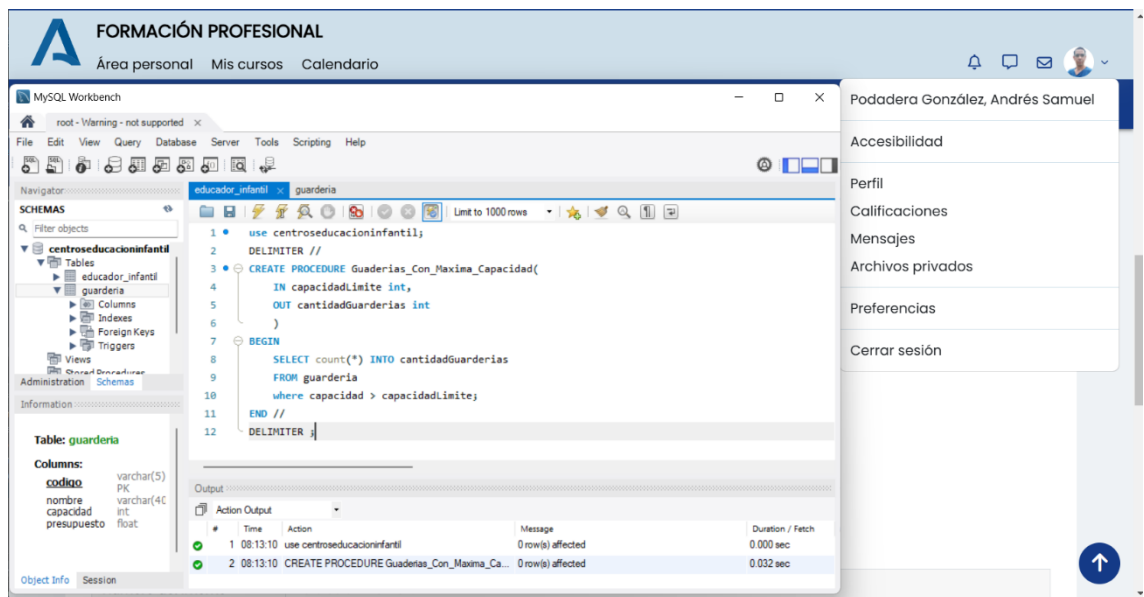
Siguiendo el mismo procedimiento que en el ejercicio anterior, conectamos el programa a la base de datos, añadiendo el conector a las librerías y comprobando que se conecta correctamente a la base de datos.

- a. **P.A. 1:** Un procedimiento almacenado que obtenga en un parámetro de salida, el número total de guarderías que superan una cierta capacidad (número máximo de

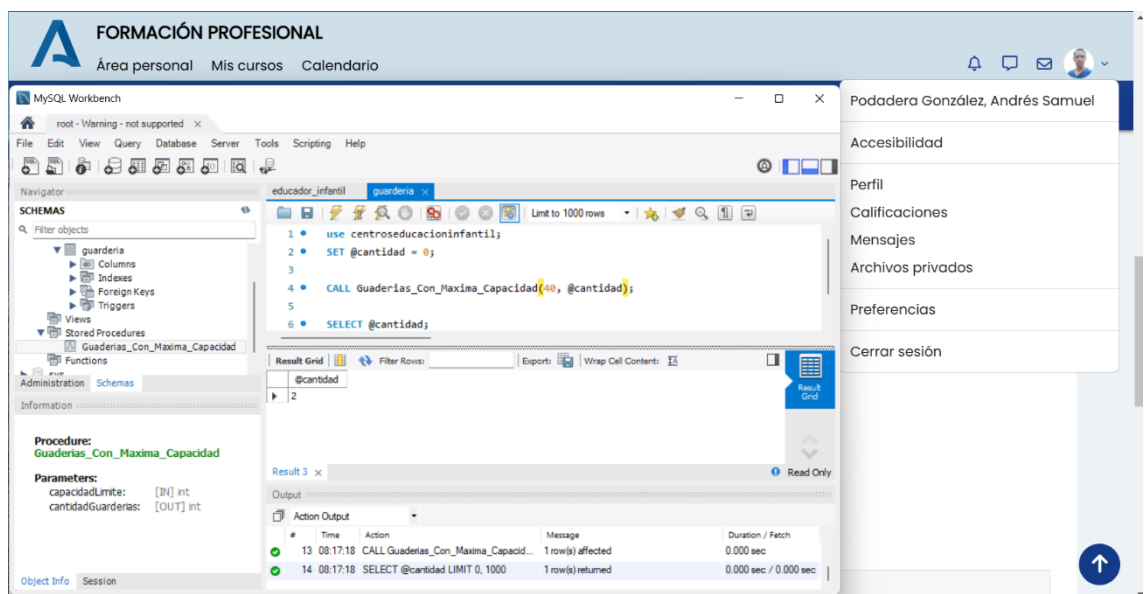
alumnos que puede acoger). Se deben pasar como parámetro al procedimiento la capacidad que se tiene que superar.

Para este primer caso detallaré el proceso que he seguido para crear el procedimiento, y que será el mismo para el resto de los procedimientos.

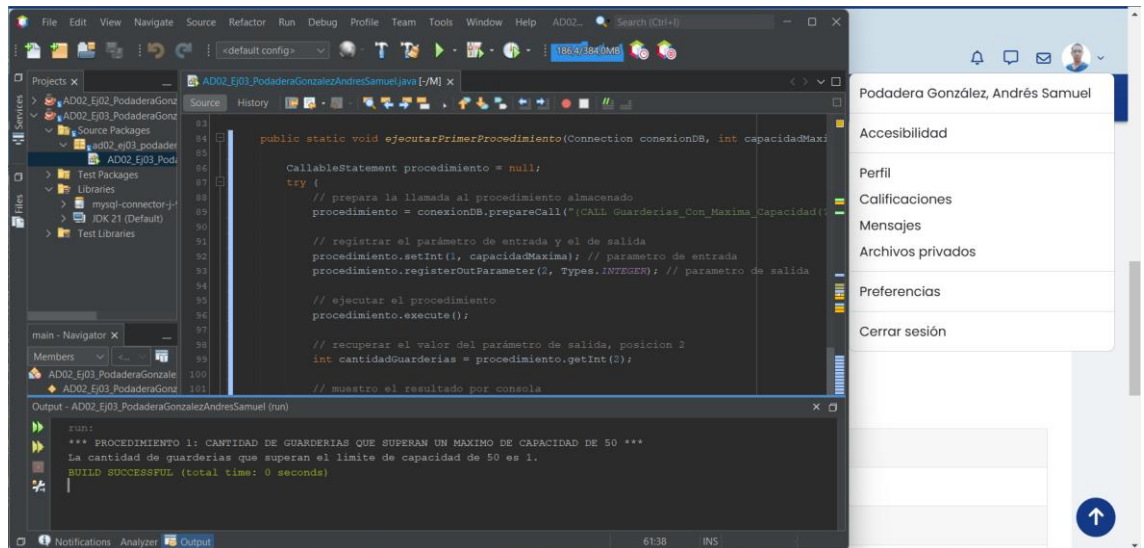
Primero usamos MySQL Workbench para crear el procedimiento y probarlo. Desde una de sus pestañas escribimos el procedimiento y lo almacenamos, haciendo clic en el símbolo del rayo.



Y comprobamos que funciona:



Ahora nos toca usarlo en Netbeans, para esto, realizamos la llamada al procedimiento y mostramos el resultado en la consola:



Código del procedimiento:

use centroseducacioninfantil;

DELIMITER //

CREATE PROCEDURE Guarderias_Con_Maxima_Capacidad(

IN capacidadLimite int,

OUT cantidadGuarderias int

)

BEGIN

SELECT count(*) INTO cantidadGuarderias

FROM guarderia

WHERE capacidad > capacidadLimite;

END //

DELIMITER ;

Pruebas en MySQL Worbench:

use centroseducacioninfantil;

SET @cantidad = 0;

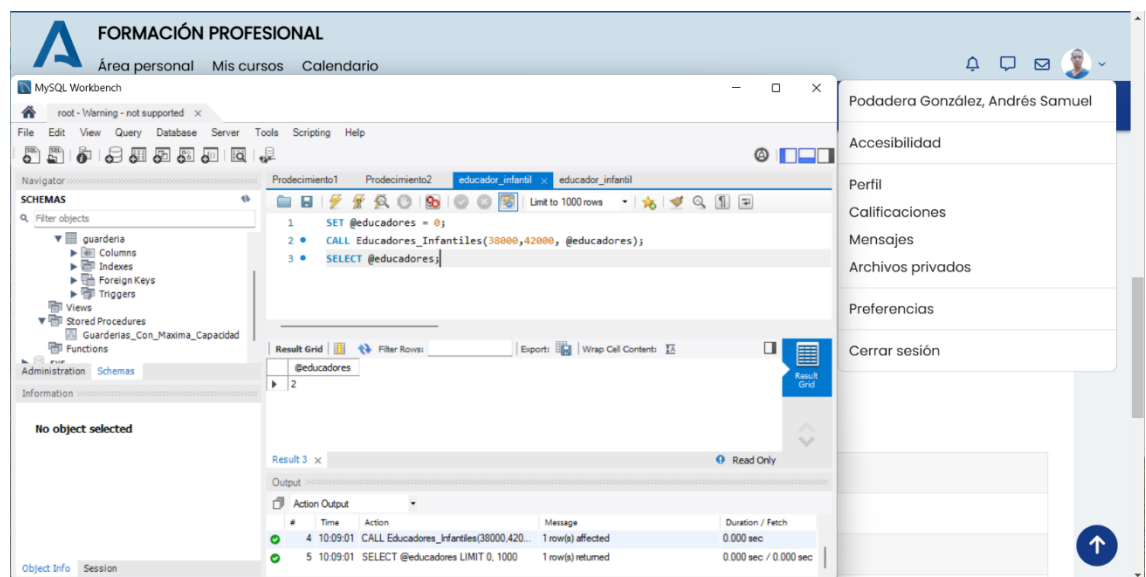
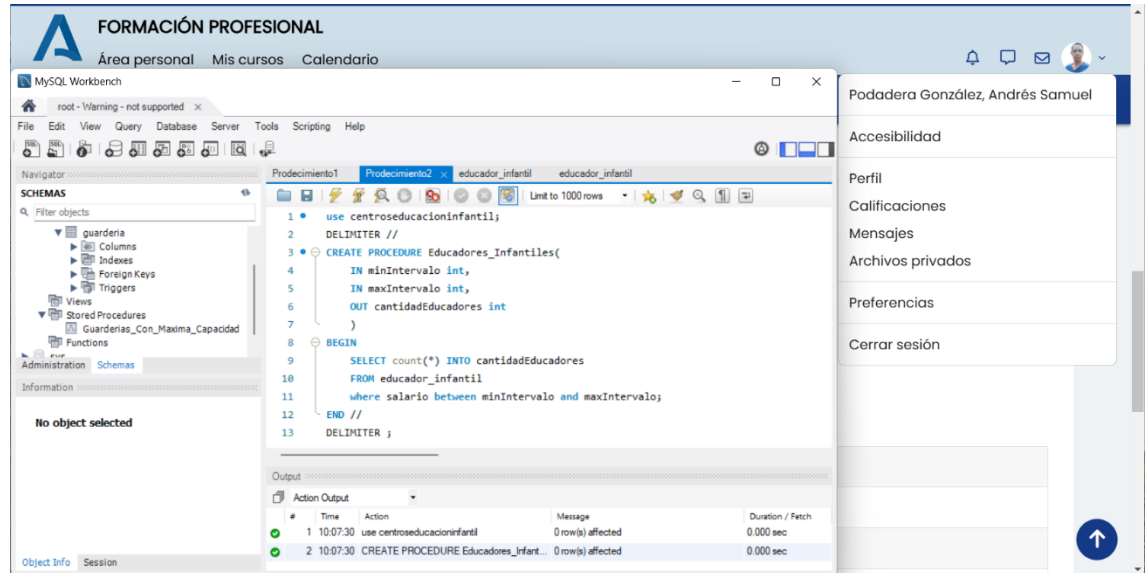
CALL Guarderias_Con_Maxima_Capacidad (40, @cantidad);

SELECT @cantidad;

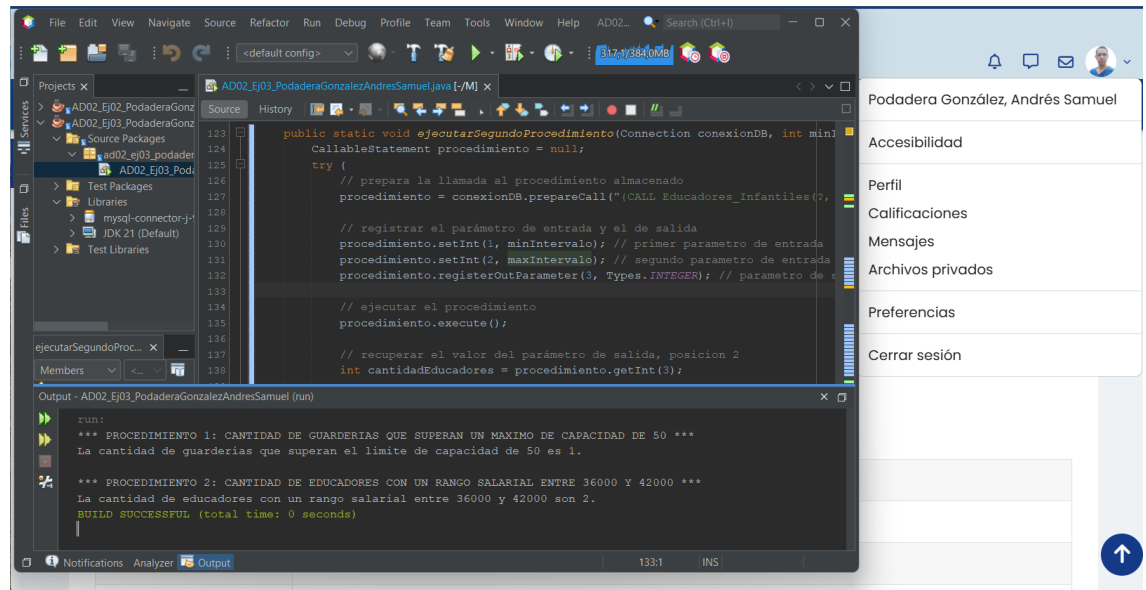
- b. **P.A. 2:** Un procedimiento almacenado que devuelva en un parámetro de salida, el número total de educadores infantiles, cuyo salario pertenece a un determinado

intervalo económico. Se deben pasar como parámetros al procedimiento, tanto el valor inicial como el valor final de dicho intervalo.

Siguiendo el mismo procedimiento del apartado anterior creamos el procedimiento y lo probamos en MySQL Workbench:



Por último, lo probamos en Netbeans:



Código del procedimiento:

use centoseducacioninfantil;

DELIMITER //

CREATE PROCEDURE Educadores_Infantiles(

IN minIntervalo int,

IN maxIntervalo int,

OUT cantidadEducadores int

)

BEGIN

SELECT count(*) INTO cantidadEducadores

FROM educador_infantil

WHERE salario BETWEEN minIntervalo AND maxIntervalo;

END //

DELIMITER ;

Pruebas en MySQL Worbench:

use centoseducacioninfantil;

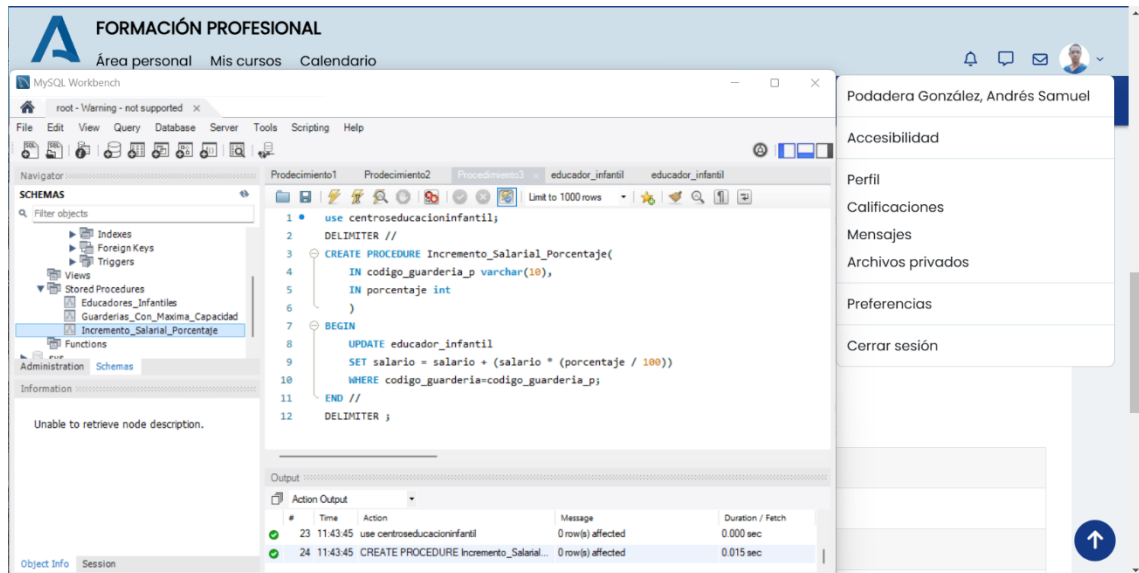
SET @educadores = 0;

CALL Educadores_Infantiles(38000,42000, @educadores);

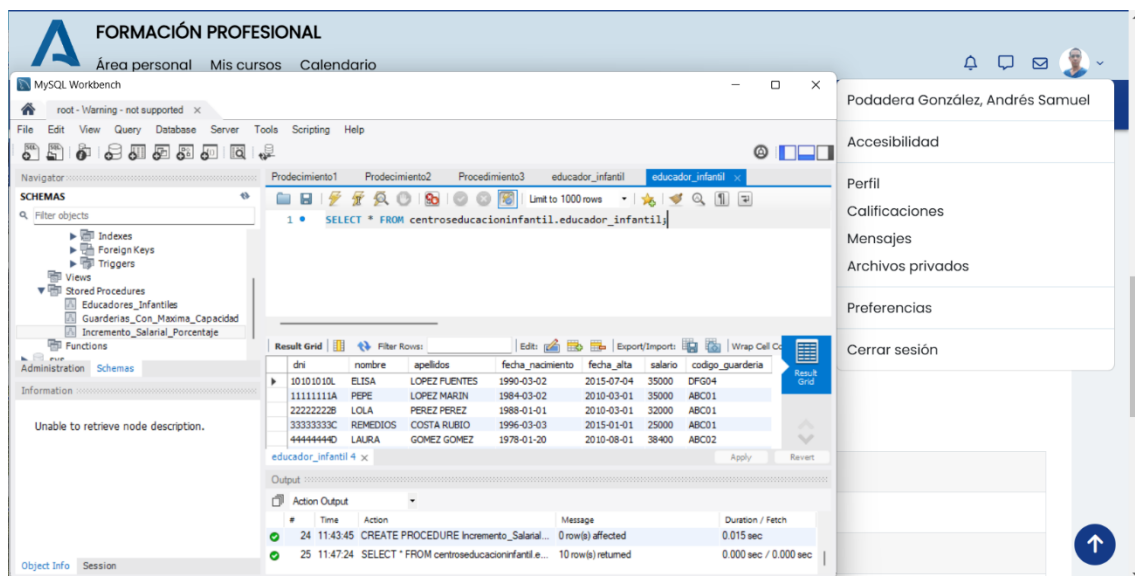
SELECT @educadores;

- c. **P.A. 3:** Un procedimiento almacenado que aumente el salario de los educadores infantiles de cierta guardería en un porcentaje. La guardería y valor entero del porcentaje a aplicar se han de pasar como parámetros (de entrada).

Siguiendo con el mismo proceso que en los apartados anteriores, creamos y probamos el procedimiento en MySQL Workbench:

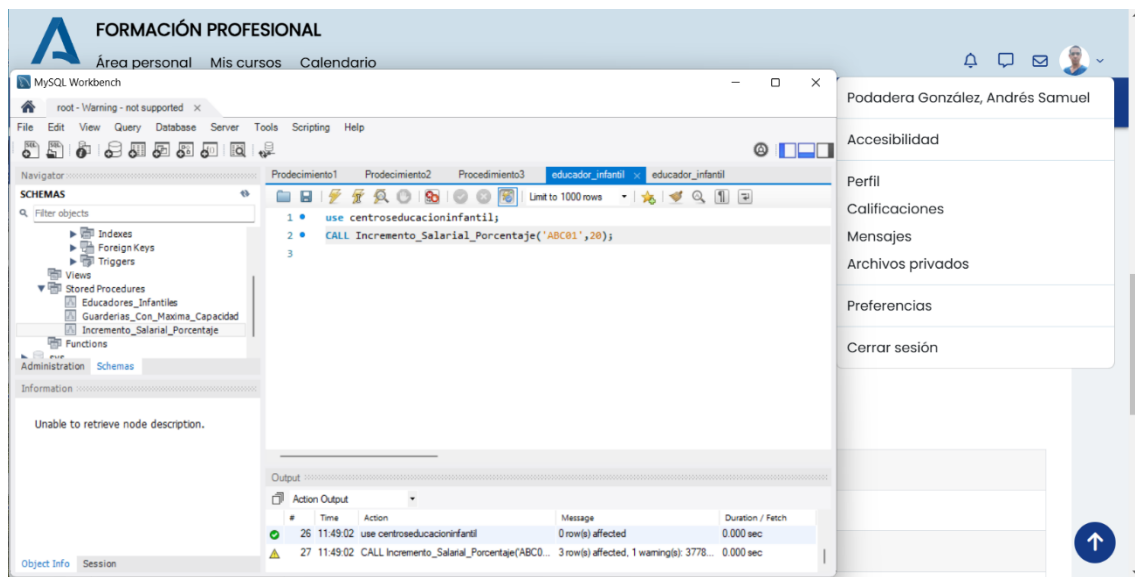


Antes de ejecutar el procedimiento:

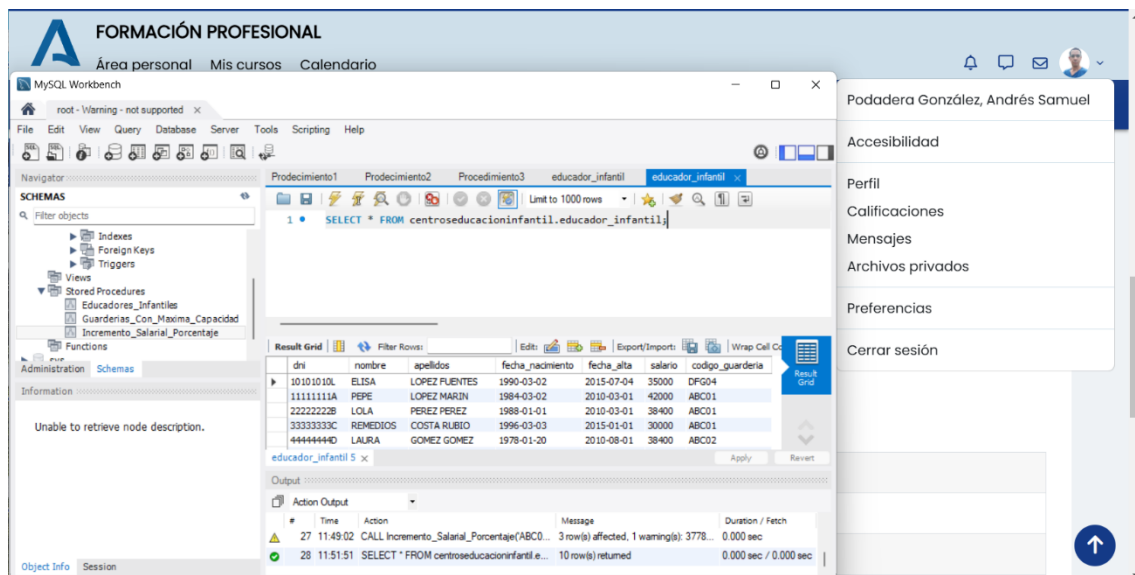


Ejecuto el procedimiento:

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

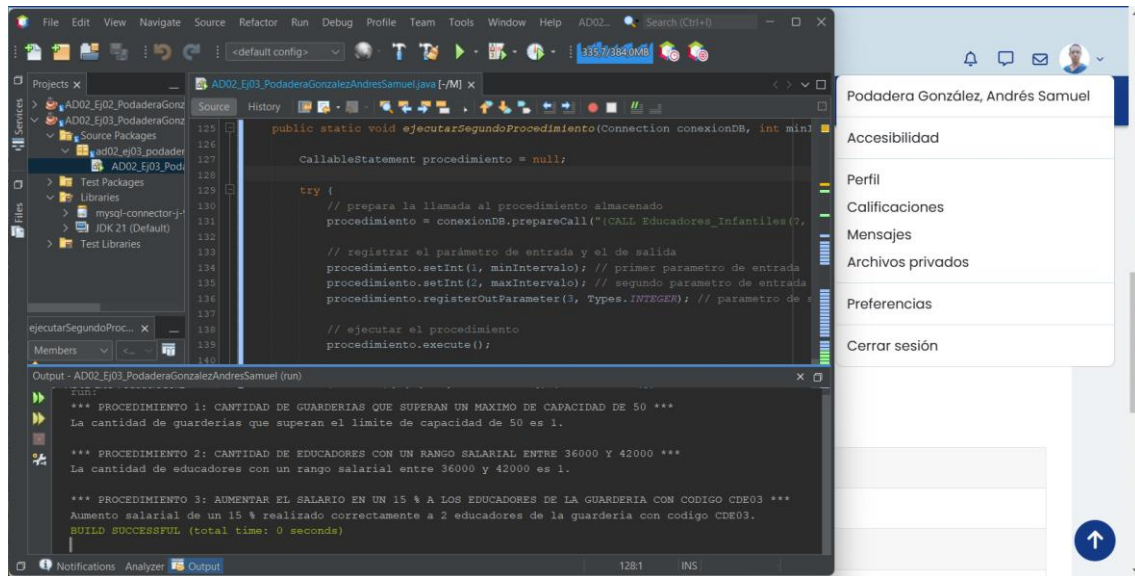


Resultado de la ejecución del procedimiento con un aumento de salario de un 20% a los educadores de la escuela con código ABC01:



Vemos que el salario a aumentado en el porcentaje indicado.

Ahora usamos el procedimiento desde nuestro programa en Java:



Código del procedimiento:

```
use centroseducacioninfantil;  
DELIMITER //  
CREATE PROCEDURE Incremento_Salarial_Porcentaje(  
    IN codigo_guarderia_p varchar(10),  
    IN porcentaje int  
)  
BEGIN  
  
    UPDATE educador_infantil  
    SET salario = salario + (salario * (porcentaje / 100))  
    WHERE codigo_guarderia=codigo_guarderia_p;  
END //  
DELIMITER ;
```

Pruebas en MySQL Workbench:

```
use centroseducacioninfantil;  
CALL Incremento_Salarial_Porcentaje('ABC01',20);
```

Debes gestionar las posibles excepciones y errores que puedan presentarse, así como el cierre de recursos utilizados de forma adecuada si la aplicación dejara de funcionar (usa la sentencia try-catch-finally o equivalente).

FIN