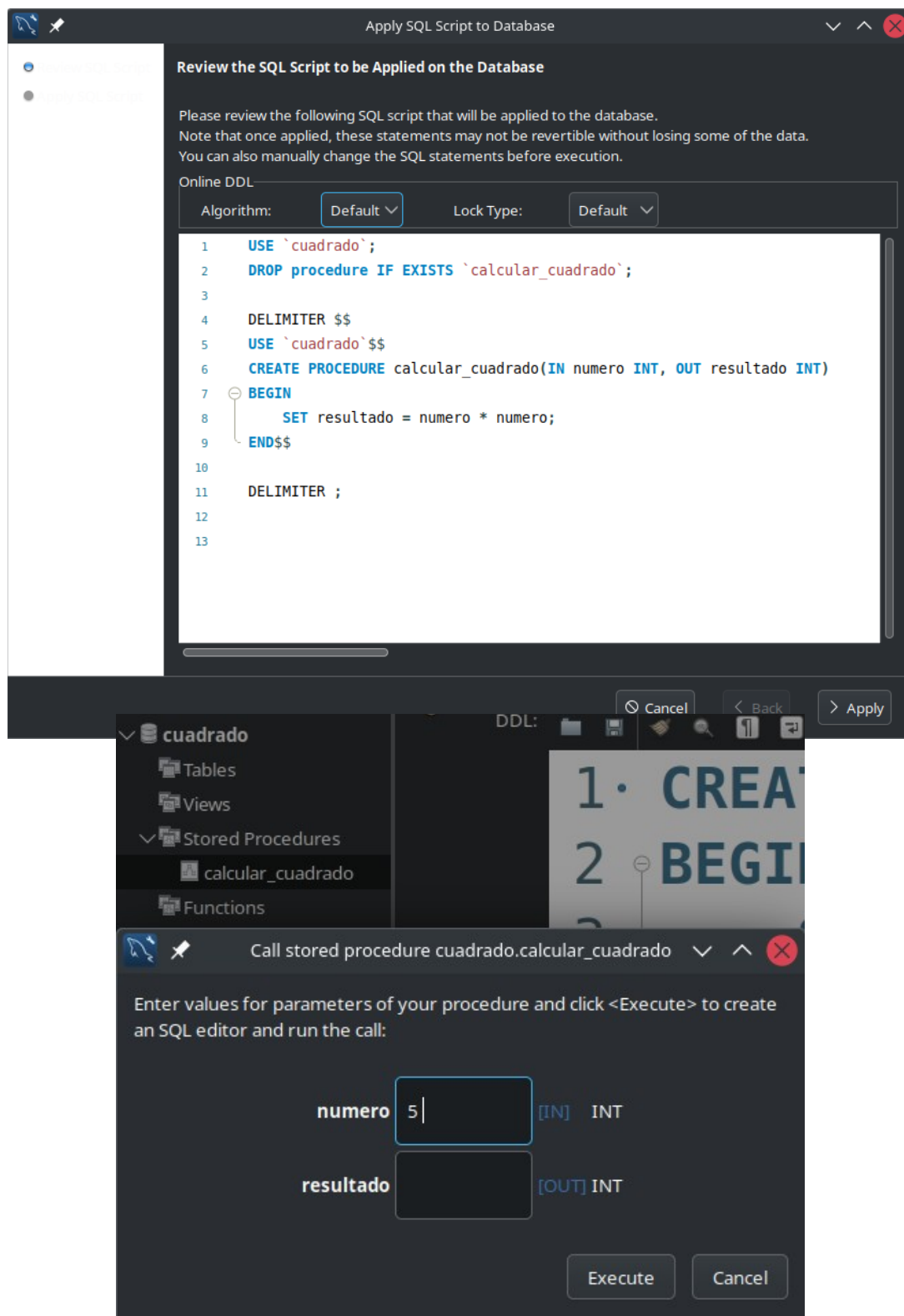


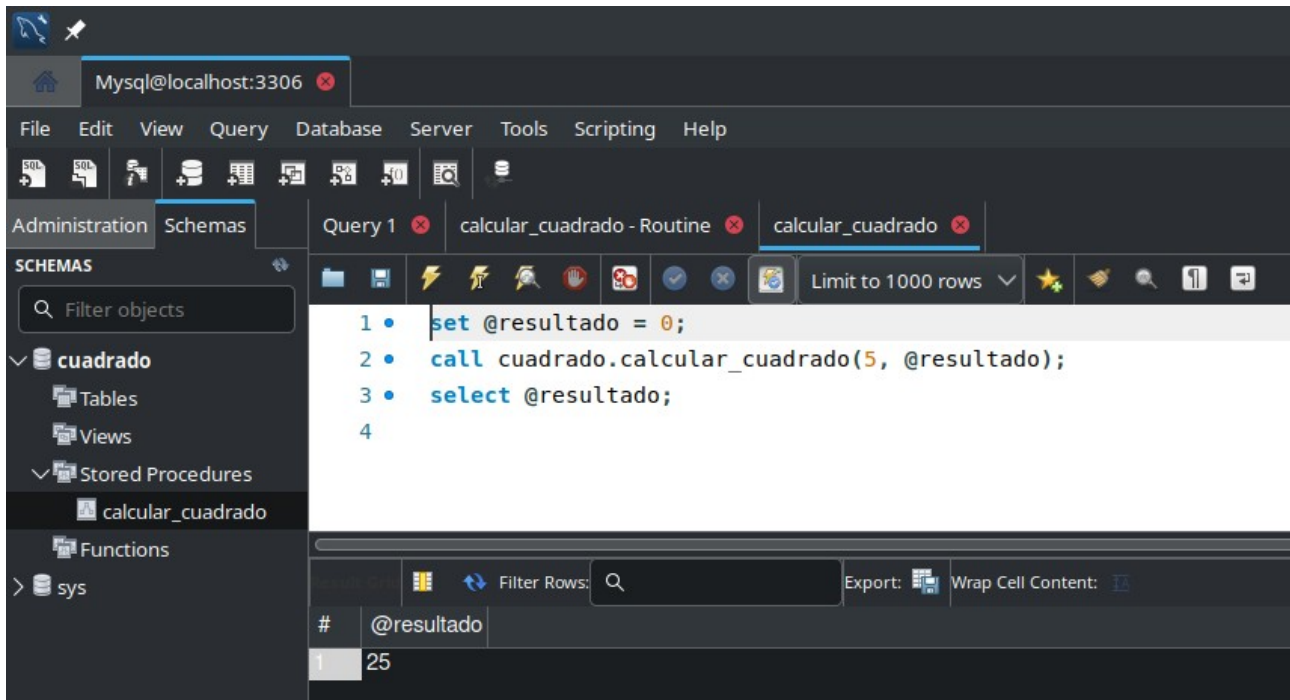
A continuación se proponen cuatro ejercicios diseñados para practicar la creación y manejo de procedimientos almacenados y funciones en MySQL Workbench.

### Ejercicio 1: Crear un Procedimiento Simple

**Objetivo:** Escribir un procedimiento que acepte un número y devuelva su cuadrado.

```
CREATE PROCEDURE `calcular_cuadrado`(IN numero INT, OUT resultado INT)
BEGIN
    SET resultado = numero * numero;
END
```





## Ejercicio 2: Procedimiento para Insertar un Nuevo Estudiante

**Objetivo:** Crear un procedimiento almacenado para insertar un nuevo estudiante en la tabla students que tiene las columnas id, name, y age.

**Crear la tabla students:**

```

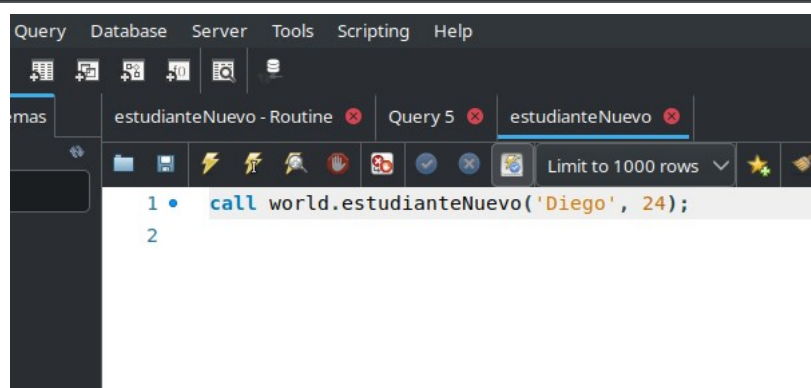
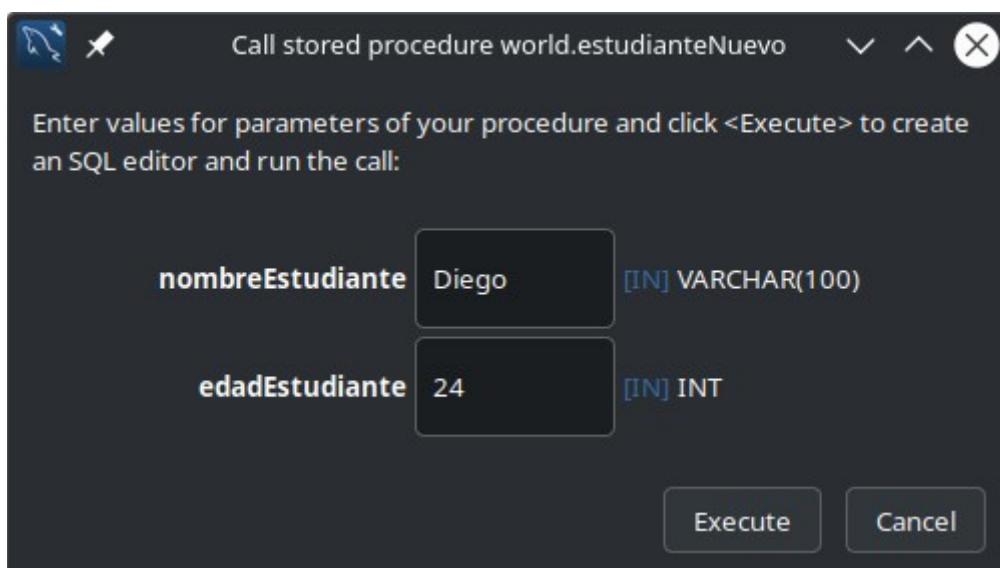
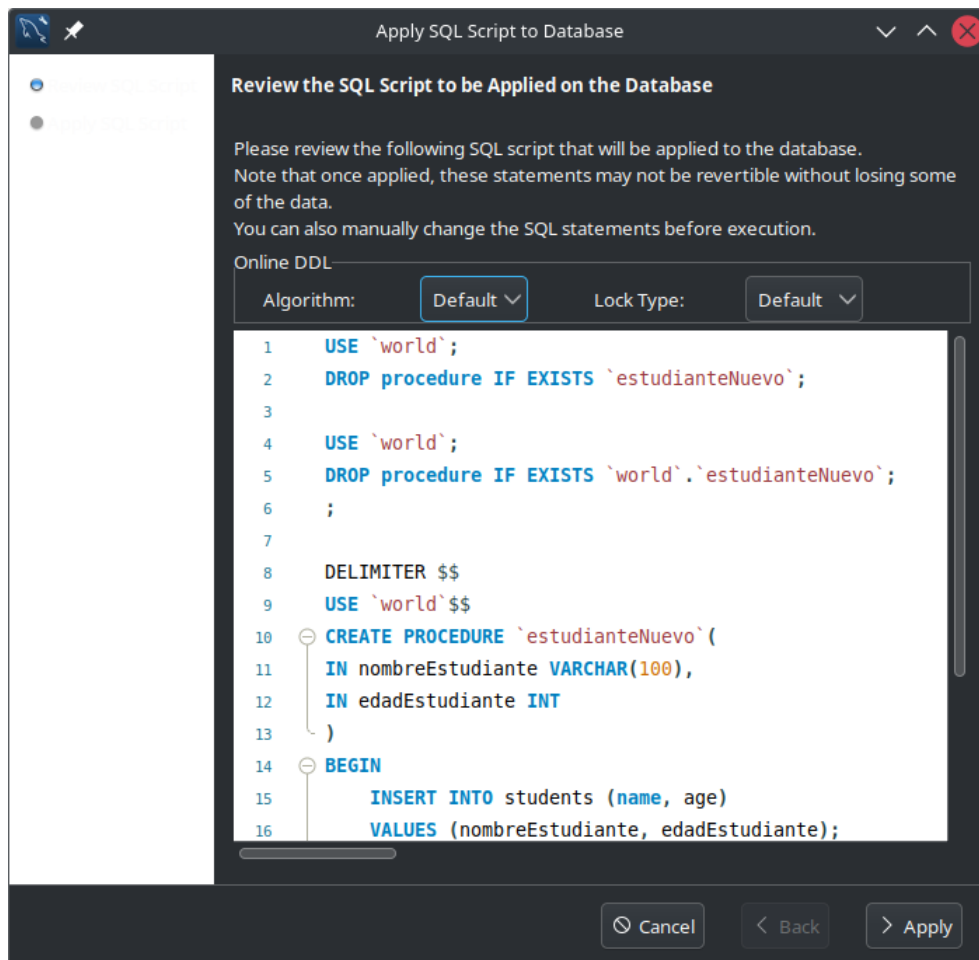
USE world;
CREATE TABLE students (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
age INT NOT NULL
);

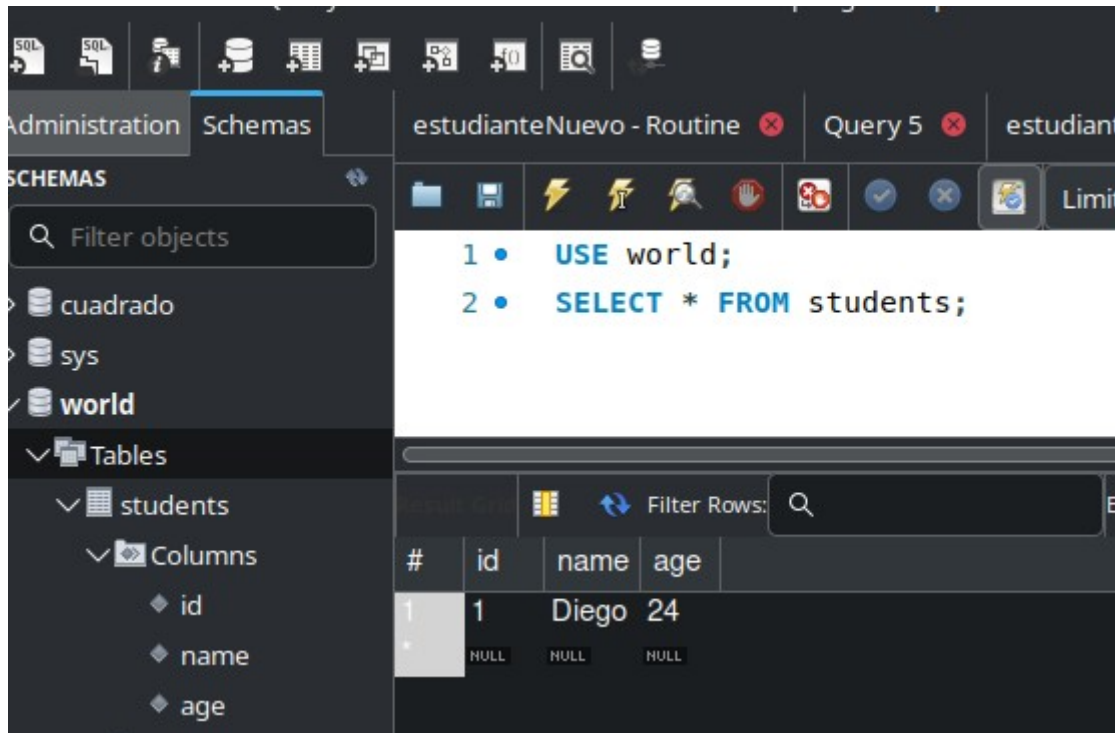
```

```

CREATE PROCEDURE `estudianteNuevo`(
IN nombreEstudiante VARCHAR(100),
IN edadEstudiante INT
)
BEGIN
INSERT INTO students (nombre, edad)
VALUES (nombreEstudiante, edadEstudiante);
END;

```

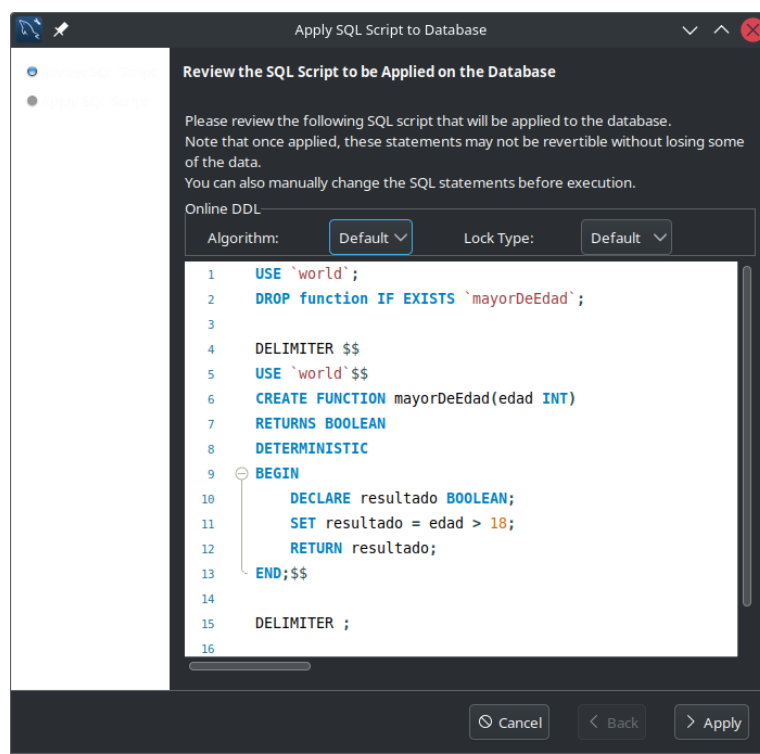




### Ejercicio 3: Función para Verificar Mayoría de Edad

**Objetivo:** Escribir una función que reciba la edad de una persona y devuelva TRUE si es mayor de 18 años, de lo contrario FALSE.

```
CREATE FUNCTION mayorDeEdad(edad INT)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
    DECLARE resultado BOOLEAN;
    SET resultado = edad > 18;
    RETURN resultado;
END;
```



Call stored function world.mayorDeEdad

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

edad 24 INT

Execute Cancel

```
1 • select world.mayorDeEdad(24);  
2
```

#	world.mayorDeEdad(24)
1	1

Call stored function world.mayorDeEdad

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

edad 16 INT

Execute Cancel

#	world.mayorDeEdad(16)
1	0

**Ejercicio 4: Procedimiento para Actualizar el Email de un Empleado**

**Objetivo:** Crear un procedimiento almacenado que actualice la dirección de correo electrónico de un empleado

específico en la tabla empleados, que contiene las columnas id, name, y email.

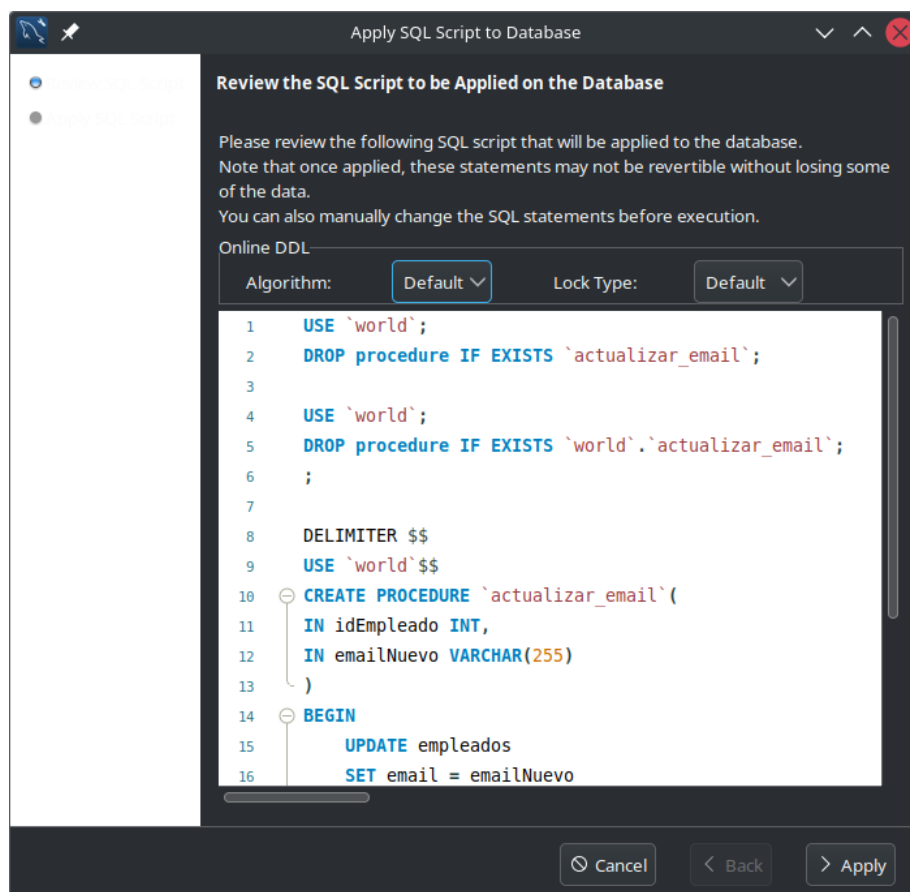
1. Crear la tabla empleados (si no existe):

```
CREATE TABLE empleados(  
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(100),  
email VARCHAR(255)  
);
```

2. Insertar datos de prueba:

```
INSERT INTO empleados (name, email) VALUES  
( 'Jose Medina', 'jmedina@mail.com'),  
( 'Ana Perez', 'aperez@mail.com');
```

```
CREATE PROCEDURE `actualizar_email`(  
IN idEmpleado INT,  
IN emailNuevo VARCHAR(255)  
)  
BEGIN  
    UPDATE empleados  
    SET email = emailNuevo  
    WHERE id = idEmpleado;  
END;
```



Call stored procedure world.actualizar\_email

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

**id\_empleado**  [IN] INT

**emailNuevo**  [IN] VARCHAR(255)

```
1 • call world.actualizar_email(1, 'email@pruebas.com');
2
```

```
1 • SELECT * FROM empleados WHERE id = 1;
2
```

#	id	name	email
1	1	Jose Medina	email@pruebas.com
		NULL	NULL
		NULL	NULL