```
17          // Establish connection
18          Connection conn = DriverManager.getConnection(url, user, password);
19          // Create PreparedStatement
20          PreparedStatement pstmt = conn.prepareStatement(insertQuery)
21      ) {
22          // Insert sample data
23          pstmt.setString(1, "ragavan");
24          pstmt.setInt(2, 20);
25          pstmt.setString(3, "A");
26          pstmt.setString(4, "ragavan.0412@gmail.com");
27
28          // Execute the query
29          int rowsInserted = pstmt.executeUpdate();
30
31          // Check if insertion was successful
32          if (rowsInserted > 0) {
33              System.out.println("A new student record was inserted successfully!");
34          }
35      } catch (SQLException e) {
36          e.printStackTrace();
37      }
38  }
```

Problems @ Javadoc Declaration Console ×

<terminated> StudentDatabase [Java Application] C:\Users\DELL\Desktop\sts-4.26.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0

A new student record was inserted successfully!

```sql
  2
  3 •    USE school;
  4
  5 • ⊖  CREATE TABLE students (
  6            id INT AUTO_INCREMENT PRIMARY KEY,
  7            name VARCHAR(100),
  8            age INT,
  9            grade VARCHAR(10),
 10            email VARCHAR(100)
 11        );
 12 •    SELECT * FROM students;
 13
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: $\overline{\underline{A}}$

Result Grid

id	name	age	grade	email
1	ragavan	20	A	ragavan.0412@gmail.com
NULL	NULL	NULL	NULL	HULL

students 3 | students 5 | students 6 ✕ | Apply | Revert

Output

Action Output

#	Time	Action	Message
✓	1 19:43:30	SELECT * FROM students LIMIT 0, 1000	3 row(s) returned
✓	2 19:43:35	SELECT * FROM students LIMIT 0, 1000	3 row(s) returned
✓	3 19:45:09	SELECT * FROM students LIMIT 0, 1000	4 row(s) returned

| JDBCCon/pom.xml | DeleteCustom... | ProductTrig... | StudentData... | *StudentDat... |

```java
15
16          try (
17              // Establishing the connection
18              Connection conn = DriverManager.getConnection(url, user, password);
19              // Creating CallableStatement
20              CallableStatement stmt = conn.prepareCall(procedureCall)
21          ) {
22              // Setting values for stored procedure parameters
23              stmt.setString(1, "ragavan");
24              stmt.setInt(2, 20);
25              stmt.setString(3, "A");
26              stmt.setString(4, "ragavan.0412@gmail.com");
27
28              // Execute the stored procedure
29              stmt.execute();
30
31              System.out.println("A new student record was inserted successfully using CallableSt
32          } catch (SQLException e) {
33              e.printStackTrace();
34          }
35      }
36  }
```

Problems   @ Javadoc   Declaration   Console ✕

<terminated> StudentDatabaseCallable [Java Application] C:\Users\DELL\Desktop\sts-4.26.0.RELEASE\plugins\org.eclipse.justj.openjdk.ho
A new student record was inserted successfully using CallableStatement!

Limit to 1000 rows ▾

```
15          IN student_name VARCHAR(100),
16          IN student_age INT,
17          IN student_grade VARCHAR(10),
18          IN student_email VARCHAR(100)
19      )
20   ⊖  BEGIN
21          INSERT INTO students (name, age, grade, email) VALUES (student_name, student_age, student_grade, studen
22      END $$
23
24      DELIMITER ;
25 ●    SELECT * FROM students;
26
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: 𝖨𝖠

id	name	age	grade	email
1	ragavan	20	A	ragavan.0412@gmail.com
NULL	NULL	NULL	NULL	NULL

Result Grid

students 3   students 5   students 6 ×   students 9   students 10   students 12   students 13   Apply   Revert   Cor

Output

Action Output ▾

#	Time	Action	Message
⊗	4 20:03:09	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCH...	Error Code: 1050. Table 'students' already exists
⊗	5 20:03:09	CREATE PROCEDURE insert_student(IN student_name VARCHAR(100), IN student_...	Error Code: 1304. PROCEDURE insert_student already exists
⊘	6 20:03:09	SELECT * FROM students LIMIT 0, 1000	2 row(s) returned