Here’s a comprehensive table that includes all stored procedures for each table in your `SchoolDB` database. Each row provides a scenario demonstrating how to use the corresponding stored procedure along with the execution code.

### Usage Scenarios for All Stored Procedures

| \*\*Stored Procedure\*\* | \*\*Scenario\*\* | \*\*Execution Code\*\* |

|--------------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------|

| \*\*InsertRole\*\* | Adding a new role 'Teacher'. | ```sql<br>EXEC InsertRole @RoleName = 'Teacher';<br>``` |

| \*\*UpdateRole\*\* | Updating role ID 1 to 'Administrator'. | ```sql<br>EXEC UpdateRole @RoleID = 1, @NewRoleName = 'Administrator';<br>``` |

| \*\*DeleteRole\*\* | Deleting role ID 2. | ```sql<br>EXEC DeleteRole @RoleID = 2;<br>``` |

| \*\*GetRoles\*\* | Retrieving all roles from the database. | ```sql<br>EXEC GetRoles;<br>``` |

| \*\*InsertUser\*\* | Adding a new user with username 'john\_doe'. | ```sql<br>EXEC InsertUser @Username = 'john\_doe', @Password = 'password123', @RoleID = 1;<br>``` |

| \*\*UpdateUser\*\* | Updating user information for user ID 1. | ```sql<br>EXEC UpdateUser @UserID = 1, @NewUsername = 'john\_updated';<br>``` |

| \*\*DeleteUser\*\* | Deleting user with ID 2. | ```sql<br>EXEC DeleteUser @UserID = 2;<br>``` |

| \*\*GetUsers\*\* | Retrieving all users from the database. | ```sql<br>EXEC GetUsers;<br>``` |

| \*\*InsertTeacher\*\* | Adding a new teacher named 'Alice Smith'. | ```sql<br>EXEC InsertTeacher @FirstName = 'Alice', @LastName = 'Smith', @Email = 'alice@example.com';<br>``` |

| \*\*UpdateTeacher\*\* | Updating teacher ID 1's email. | ```sql<br>EXEC UpdateTeacher @TeacherID = 1, @NewEmail = 'alice\_updated@example.com';<br>``` |

| \*\*DeleteTeacher\*\* | Deleting teacher with ID 2. | ```sql<br>EXEC DeleteTeacher @TeacherID = 2;<br>``` |

| \*\*GetTeachers\*\* | Retrieving all teachers from the database. | ```sql<br>EXEC GetTeachers;<br>``` |

| \*\*InsertStudent\*\* | Adding a new student named 'Bob Johnson'. | ```sql<br>EXEC InsertStudent @FirstName = 'Bob', @LastName = 'Johnson', @DateOfBirth = '2005-04-01', @Email = 'bob@example.com';<br>``` |

| \*\*UpdateStudent\*\* | Updating student ID 1's last name. | ```sql<br>EXEC UpdateStudent @StudentID = 1, @NewLastName = 'Johnson-Smith';<br>``` |

| \*\*DeleteStudent\*\* | Deleting student with ID 2. | ```sql<br>EXEC DeleteStudent @StudentID = 2;<br>``` |

| \*\*GetStudents\*\* | Retrieving all students from the database. | ```sql<br>EXEC GetStudents;<br>``` |

| \*\*InsertContact\*\* | Adding a new contact for student ID 1. | ```sql<br>EXEC InsertContact @StudentID = 1, @Phone = '123-456-7890';<br>``` |

| \*\*UpdateContact\*\* | Updating contact ID 1's phone number. | ```sql<br>EXEC UpdateContact @ContactID = 1, @NewPhone = '987-654-3210';<br>``` |

| \*\*DeleteContact\*\* | Deleting contact with ID 2. | ```sql<br>EXEC DeleteContact @ContactID = 2;<br>``` |

| \*\*GetContacts\*\* | Retrieving all contacts from the database. | ```sql<br>EXEC GetContacts;<br>``` |

| \*\*InsertClass\*\* | Creating a new class called 'Biology 101'. | ```sql<br>EXEC InsertClass @ClassName = 'Biology 101', @TeacherID = 1;<br>``` |

| \*\*UpdateClass\*\* | Renaming class with ID 1 to 'Advanced Biology'. | ```sql<br>EXEC UpdateClass @ClassID = 1, @NewClassName = 'Advanced Biology';<br>``` |

| \*\*DeleteClass\*\* | Deleting class with ID 2. | ```sql<br>EXEC DeleteClass @ClassID = 2;<br>``` |

| \*\*GetClasses\*\* | Getting a list of all classes. | ```sql<br>EXEC GetClasses;<br>``` |

| \*\*InsertSubject\*\* | Adding a new subject named 'Mathematics'. | ```sql<br>EXEC InsertSubject @SubjectName = 'Mathematics';<br>``` |

| \*\*UpdateSubject\*\* | Updating subject name for ID 1 to 'Algebra'. | ```sql<br>EXEC UpdateSubject @SubjectID = 1, @NewSubjectName = 'Algebra';<br>``` |

| \*\*DeleteSubject\*\* | Deleting subject with ID 2. | ```sql<br>EXEC DeleteSubject @SubjectID = 2;<br>``` |

| \*\*GetSubjects\*\* | Retrieving all subjects from the database. | ```sql<br>EXEC GetSubjects;<br>``` |

| \*\*InsertClassSubject\*\* | Linking class ID 1 to subject ID 1. | ```sql<br>EXEC InsertClassSubject @ClassID = 1, @SubjectID = 1;<br>``` |

| \*\*DeleteClassSubject\*\* | Removing the link between class subject ID 1. | ```sql<br>EXEC DeleteClassSubject @ClassSubjectID = 1;<br>``` |

| \*\*GetClassSubjects\*\* | Getting all class-subject relationships. | ```sql<br>EXEC GetClassSubjects;<br>``` |

| \*\*InsertEnrollment\*\* | Enrolling student ID 1 in class ID 1. | ```sql<br>EXEC InsertEnrollment @StudentID = 1, @ClassID = 1, @EnrollmentDate = GETDATE();<br>``` |

| \*\*DeleteEnrollment\*\* | Unenrolling enrollment ID 1. | ```sql<br>EXEC DeleteEnrollment @EnrollmentID = 1;<br>``` |

| \*\*GetEnrollments\*\* | Retrieving all enrollments from the database. | ```sql<br>EXEC GetEnrollments;<br>``` |

| \*\*InsertGrade\*\* | Assigning a grade of 85.5 for student ID 1 in class ID 1. | ```sql<br>EXEC InsertGrade @StudentID = 1, @ClassSubjectID = 1, @Grade = 85.5, @GradeDate = GETDATE();<br>``` |

| \*\*UpdateGrade\*\* | Updating grade ID 1 to a new value of 90. | ```sql<br>EXEC UpdateGrade @GradeID = 1, @NewGrade = 90;<br>``` |

| \*\*DeleteGrade\*\* | Deleting grade entry ID 2. | ```sql<br>EXEC DeleteGrade @GradeID = 2;<br>``` |

| \*\*GetGrades\*\* | Retrieving all grades from the database. | ```sql<br>EXEC GetGrades;<br>``` |

### Explanation of Usage

1. \*\*Insert Procedures\*\*: Used to add new records to the respective tables, such as adding a new role, user, teacher, student, etc.

2. \*\*Update Procedures\*\*: Modify existing records by updating specific fields, like changing a user's username or a class's name.

3. \*\*Delete Procedures\*\*: Remove records from the tables, such as deleting a user, teacher, or enrollment.

4. \*\*Get Procedures\*\*: Retrieve data from the database to view all records from the respective tables, like users, classes, subjects, etc.

### Conclusion

This table provides practical examples of how to utilize your stored procedures in `SchoolDB`. By executing the provided SQL code, you can effectively manage and manipulate data within your school database. Feel free to modify the values as needed for your specific use cases.