# Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology, Pune-37

*(An autonomous Institute of Savitribai Phule Pune University)*

**Department of Multidisciplinary Engineering**

|  |  |
| --- | --- |
| **Roll. No** | 58 |
| **Name** | Aaditya M Patil |
| **Division** | CS-C |
| **Batch** | 3 |
| **GR. No** | 12210643 |

**Assignment No. 6**

**Title:** Write useful stored procedures in PL/SQL.

**Description:**

* **Procedure in MySQL:** A procedure in MySQL is a named collection of SQL statements that can be stored in the database and executed as a single unit. Procedures are often used to encapsulate a sequence of SQL commands and provide a way to reuse and modularize database logic. They can accept parameters, which makes them flexible and adaptable to various scenarios.
* **Creating a Procedure:**
* To create a procedure in MySQL, you can use the following syntax:
* CREATE PROCEDURE procedure\_name ([parameter\_list])
* BEGIN

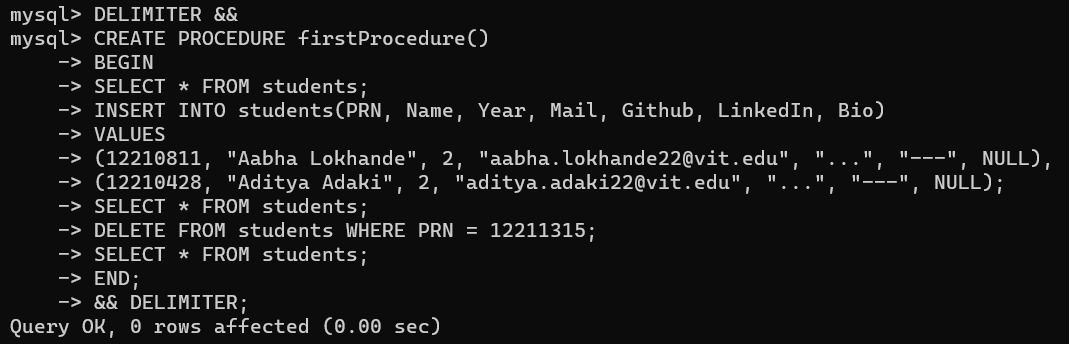
--- SQL statements

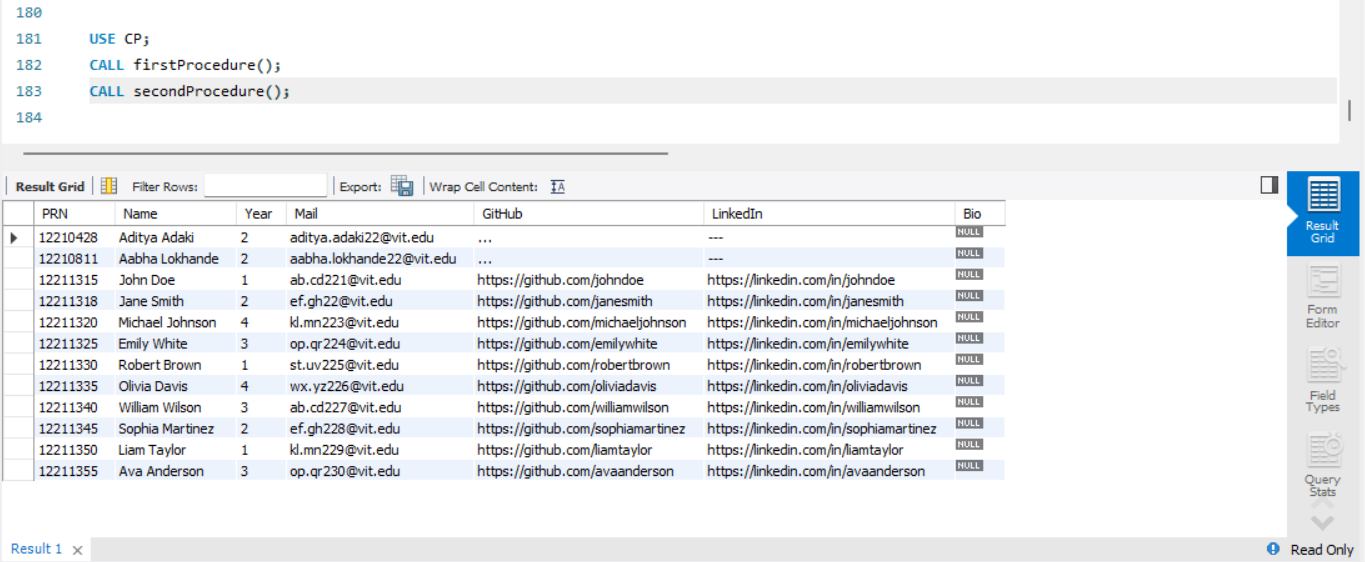
* END;
* Inside the BEGIN and END block, you can define the SQL statements that make up the procedure's logic.
* **Parameters:**
* Parameters are placeholders for values that can be passed to a procedure when it's called. They are defined in the parameter list as follows:
* parameter\_name datatype
* Parameters make it possible to pass data into and out of the procedure, allowing for dynamic and flexible behavior.
* **Executing a Procedure:**
* To execute a procedure in MySQL, you can use the CALL statement with the procedure name and provide the necessary parameter values if the procedure expects them:
* CALL procedure\_name([parameter\_values]);
* Executing a procedure runs the SQL statements defined within it.
* **Deleting a Procedure:**
* If you no longer need a procedure, you can delete it using the DROP PROCEDURE statement:
* DROP PROCEDURE procedure\_name;
* This removes the procedure from the database, and it can no longer be executed.
* These elements together enable you to create, execute, and manage procedures in MySQL, providing a powerful way to encapsulate and reuse database logic.

**Screenshots/Output:**

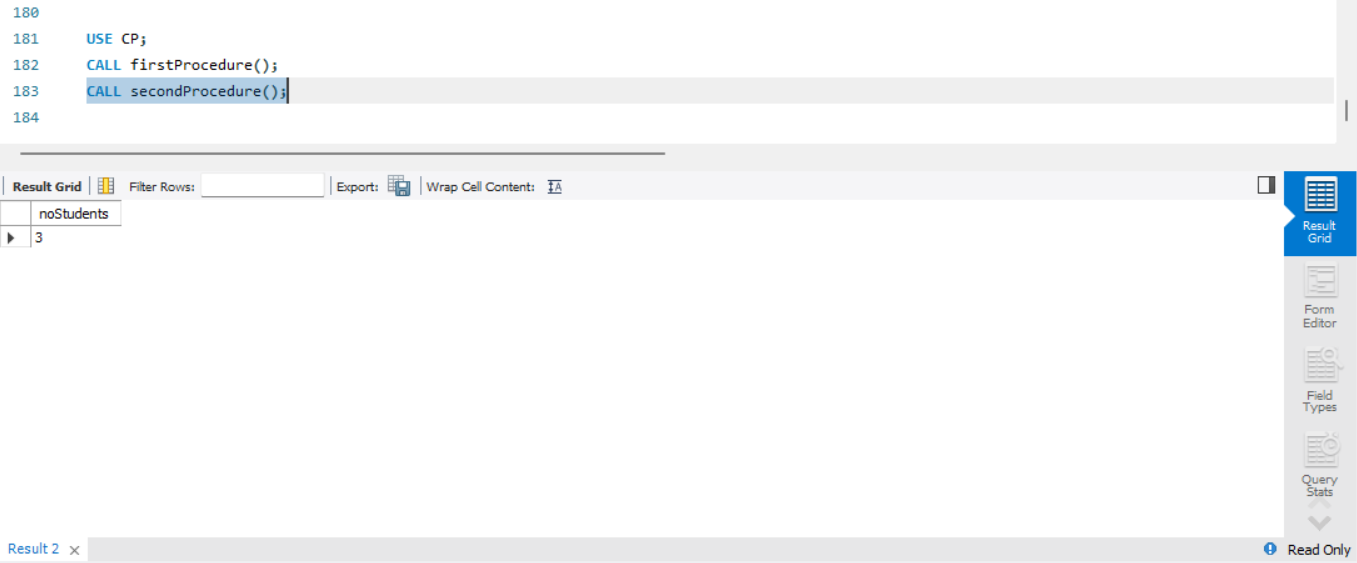
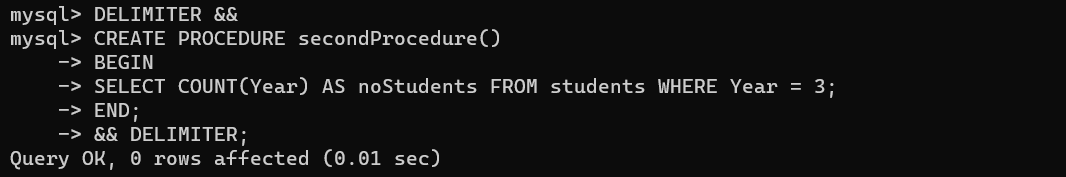
students table:



First Procedure:  




Second Procedure:



**Conclusion:**

In conclusion, stored procedures in MySQL are valuable for encapsulating and reusing database logic. They allow you to create modular and flexible code that can be executed as a single unit. This assignment demonstrated the creation and use of two simple PL/SQL procedures: one for retrieving student information by PRN and another for updating a student's information by PRN. These procedures can be customized and expanded to suit various database-related tasks.