

With respect to the table employee and department created, Create a stored procedure named as CountEmployee that returns the count of total number of employee in employee Table.

```
DELIMITER //
```

```
CREATE PROCEDURE CountEmployee(OUT emp_count INT)
```

```
BEGIN
```

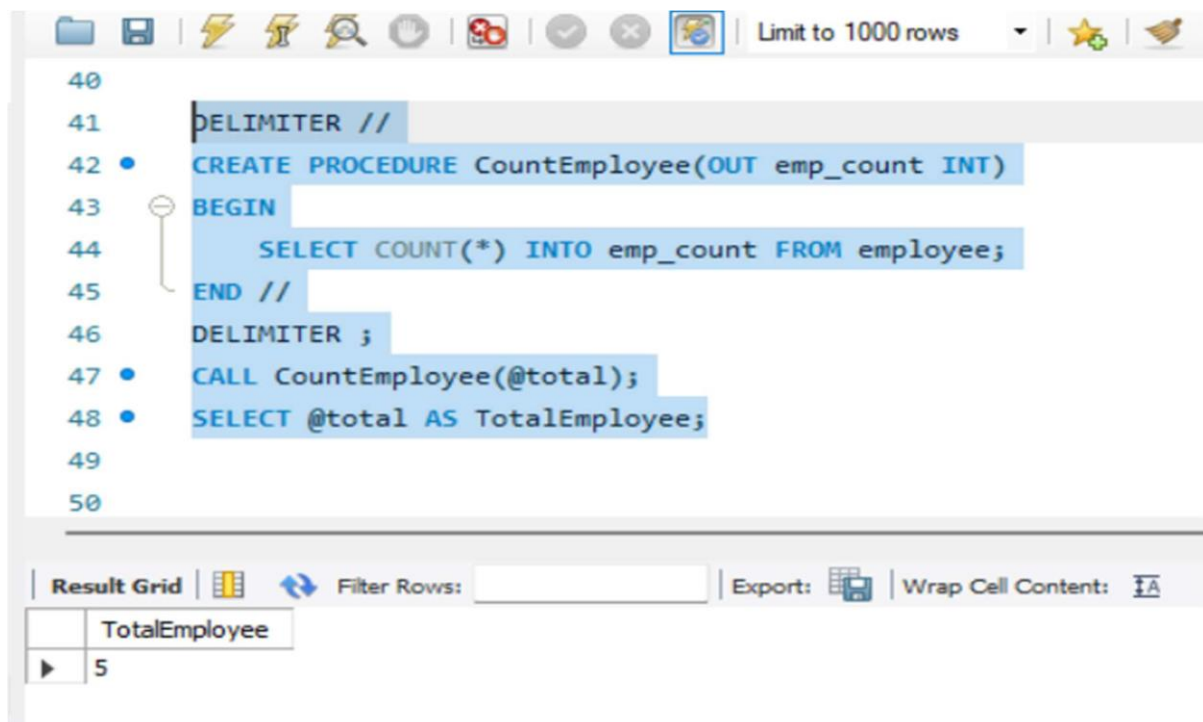
```
SELECT COUNT(*) INTO emp_count FROM employee;
```

```
END //
```

```
DELIMITER ;
```

```
CALL CountEmployee(@total);
```

```
SELECT @total AS TotalEmployee;
```



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The main editor displays SQL code with line numbers 40 to 50. Lines 41 through 48 are highlighted in blue. The code defines a stored procedure 'CountEmployee' that takes an output parameter 'emp_count' and returns the count of rows in the 'employee' table. It then calls this procedure with '@total' and selects the result as 'TotalEmployee'.

```
40
41 DELIMITER //
42 CREATE PROCEDURE CountEmployee(OUT emp_count INT)
43 BEGIN
44     SELECT COUNT(*) INTO emp_count FROM employee;
45 END //
46 DELIMITER ;
47 CALL CountEmployee(@total);
48 SELECT @total AS TotalEmployee;
49
50
```

Below the editor, the 'Result Grid' tab is active. It shows a single row with the column 'TotalEmployee' and the value '5'.

TotalEmployee
5