

## Stored Procedure to Add a New Record to the Worker Table and called the procedure

The screenshot shows the MySQL Workbench interface with the 'worker\_db' schema selected. The SQL editor contains the following code:

```
12
13 BEGIN
14     INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)
15     VALUES (p_Worker_Id, p_FirstName, p_LastName, p_Salary, p_JoiningDate, p_Department);
16 END $$
17
18 DELIMITER ;
19
20 -- Procedure call example
21 CALL AddWorker(12, 'Peter', 'Sam', 57000, '2024-11-12 09:00:00', 'Engineering');
22 select * from Worker;
```

The 'Result Grid' displays the current state of the 'Worker' table:

Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
1	John	Doe	55000	2023-01-15 08:30:00	Finance
2	Jane	Smith	65000	2022-03-10 09:00:00	Finance
3	Mary	Johnson	70000	2021-07-25 10:00:00	IT
4	James	Brown	60000	2020-09-05 11:30:00	Marketing
5	Patricia	Davis	75000	2021-05-12 08:45:00	Sales
6	Michael	Miller	80000	2020-11-22 09:15:00	HR
7	Linda	Wilson	67000	2022-04-18 10:30:00	Finance
8	Robert	Moore	72000	2019-08-03 07:30:00	IT
9	Elizabeth	Taylor	85000	2018-12-19 11:00:00	Operations
10	William	Anderson	90000	2020-06-20 08:00:00	Sales
11	Abraham	Kureshi	85000	2024-11-12 10:00:00	IT
12	Peter	Sam	57000	2024-11-12 09:00:00	Engineering

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	21:51:37	CREATE PROCEDURE AddWorker( IN p_Worker_Id INT, IN p_FirstName CHAR(25), IN p_L...	0 row(s) affected	0.000 sec
2	21:52:02	select * from	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your M...	0.000 sec
3	21:52:08	select * from worker LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
4	21:52:43	CALL AddWorker(12, 'Peter', 'Sam', 57000, '2024-11-12 09:00:00', 'Engineering')	1 row(s) affected	0.015 sec
5	21:53:02	select * from Worker LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec

## Stored Procedure to Retrieve Salary of a Worker by ID and called the procedure

The screenshot shows the MySQL Workbench interface with the 'worker\_db' schema selected. The SQL editor contains the following code:

```
32 BEGIN
33     SELECT Salary INTO p_Salary
34     FROM Worker
35     WHERE Worker_Id = p_Worker_Id;
36 END $$
37
38 DELIMITER ;
39
40 -- Procedure call example
41 CALL GetSalaryByWorkerId(11, @v_Salary);
42 SELECT @v_Salary AS Salary;
```

The 'Result Grid' displays the output of the procedure call:

Salary
85000

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
6	21:53:45	CREATE PROCEDURE GetSalaryByWorkerId( IN p_Worker_Id INT, OUT p_Salary INT ) BEG...	0 row(s) affected	0.016 sec
7	21:54:44	CALL GetSalaryByWorkerId(101, @v_Salary)	0 row(s) affected, 1 warning(s): 1329 No data - zero rows fetched, selected, or processed	0.000 sec
8	21:54:47	SELECT @v_Salary AS Salary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
9	21:55:08	DECLARE v_Salary INT	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your ...	0.000 sec
10	21:55:35	CALL GetSalaryByWorkerId(11, @v_Salary)	1 row(s) affected	0.016 sec
11	21:55:39	SELECT @v_Salary AS Salary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## Stored Procedure to Update Department of a Worker by Worker ID and called the procedure

The screenshot shows the MySQL Workbench interface with the 'worker\_db' schema selected. The SQL editor contains the following code:

```
-- Stored Procedure to Update Department of a Worker by Worker ID
DELIMITER $$
CREATE PROCEDURE UpdateWorkerDepartment(
    IN p_Worker_Id INT,
    IN p_Department CHAR(25)
)
BEGIN
    UPDATE Worker
    SET Department = p_Department
    WHERE Worker_Id = p_Worker_Id;
END $$
DELIMITER ;

-- Procedure call example
CALL UpdateWorkerDepartment(6, 'HR');
```

The Output window shows the execution results:

#	Time	Action	Message	Duration / Fetch
11	21:55:39	SELECT @v_Salary AS Salary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
12	21:56:06	CREATE PROCEDURE UpdateWorkerDepartment( IN p_Worker_Id INT, IN p_Department CH...	0 row(s) affected	0.032 sec
13	21:56:14	select * from worker LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
14	21:56:21	CALL UpdateWorkerDepartment(6, 'HR')	0 row(s) affected	0.000 sec
15	21:57:03	select * from worker LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
16	21:57:11	CALL UpdateWorkerDepartment(6, 'HR')	0 row(s) affected	0.000 sec

## Stored Procedure to Retrieve the Number of Workers in a Given Department and called procedure

The screenshot shows the MySQL Workbench interface with the 'worker\_db' schema selected. The SQL editor contains the following code:

```
BEGIN
SELECT COUNT(*) INTO p_WorkerCount
FROM Worker
WHERE Department = p_Department;
END $$
DELIMITER ;

-- Procedure call example
CALL GetWorkerCountByDepartment('Engineering', @v_WorkerCount);
SELECT @v_WorkerCount AS WorkerCount;
```

The Output window shows the execution results:

#	Time	Action	Message	Duration / Fetch
16	21:57:11	CALL UpdateWorkerDepartment(6, 'HR')	0 row(s) affected	0.000 sec
17	21:58:00	CREATE PROCEDURE GetWorkerCountByDepartment( IN p_Department CHAR(25), OUT p_...	0 row(s) affected	0.031 sec
18	21:58:32	CALL GetWorkerCountByDepartment('Engineering', v_WorkerCount)	Error Code: 1054. Unknown column 'v_WorkerCount' in field list	0.000 sec
19	21:58:46	CALL GetWorkerCountByDepartment('Engineering', p_WorkerCount)	Error Code: 1054. Unknown column 'p_WorkerCount' in field list	0.000 sec
20	21:59:27	CALL GetWorkerCountByDepartment('Engineering', @v_WorkerCount)	1 row(s) affected	0.000 sec
21	21:59:28	SELECT @v_WorkerCount AS WorkerCount LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The Result Grid shows the output of the final SELECT statement:

WorkerCount
1

## Stored Procedure to Retrieve the Average Salary of Workers in a Given Department

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar shows the 'SCHEMAS' panel with a tree view of the 'worker\_db' database, including tables, views, stored procedures, and functions. The main editor window shows a SQL script for a stored procedure named 'GetAvgSalaryByDepartment'. The script defines a procedure that takes a department name and an output variable for average salary, calculates the average salary from the 'Worker' table, and returns it. Below the script, the 'Result Grid' shows the output of the procedure call, displaying a single row with the average salary of 57000.00. The bottom panel shows the 'Output' tab with a table of execution messages, including the procedure call, the SELECT statement, and the return value.

```
SQL File 4*  SQL File 4*  SQL File 6*  SQL File 6* x
Limit to 1000 rows
91 BEGIN
92     SELECT AVG(Salary) INTO p_AvgSalary
93     FROM Worker
94     WHERE Department = p_Department;
95 END $$
96
97 DELIMITER ;
98
99 -- Procedure call example
100 CALL GetAvgSalaryByDepartment('Engineering', @v_AvgSalary);
101 SELECT @v_AvgSalary AS AvgSalary;
```

AvgSalary	
57000.00	

Schema: worker\_db

#	Time	Action	Message	Duration / Fetch
20	21:59:27	CALL GetWorkerCountByDepartment('Engineering', @v_WorkerCount)	1 row(s) affected	0.000 sec
21	21:59:28	SELECT @v_WorkerCount AS WorkerCount LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
22	21:59:55	CALL UpdateWorkerDepartment(5, 'HR')	1 row(s) affected	0.015 sec
23	22:00:14	CREATE PROCEDURE GetAvgSalaryByDepartment( IN p_Department CHAR(25), OUT p_Avg...	0 row(s) affected	0.000 sec
24	22:00:17	CALL GetAvgSalaryByDepartment('Engineering', @v_AvgSalary)	1 row(s) affected	0.000 sec
25	22:00:19	SELECT @v_AvgSalary AS AvgSalary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

