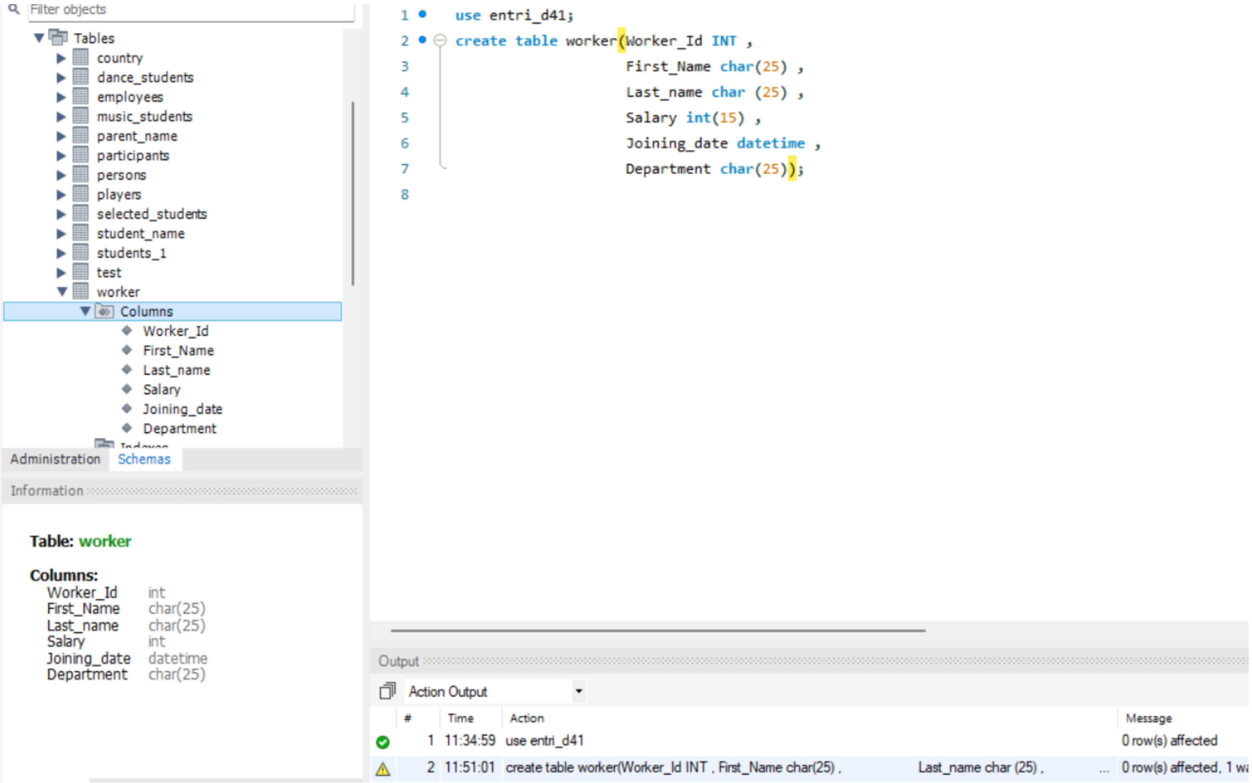
Stored Procedures

Consider the Worker table with following fields: Worker_Id INT FirstName CHAR(25), LastName CHAR(25), Salary INT(15), JoiningDate DATETIME, Department CHAR(25))



```
use entri_d41;
create table worker(Worker_Id INT ,
                    First_Name char(25) ,
                    Last_name char (25) ,
                    Salary int(15) ,
                    Joining_date datetime ,
                    Department char(25));
```

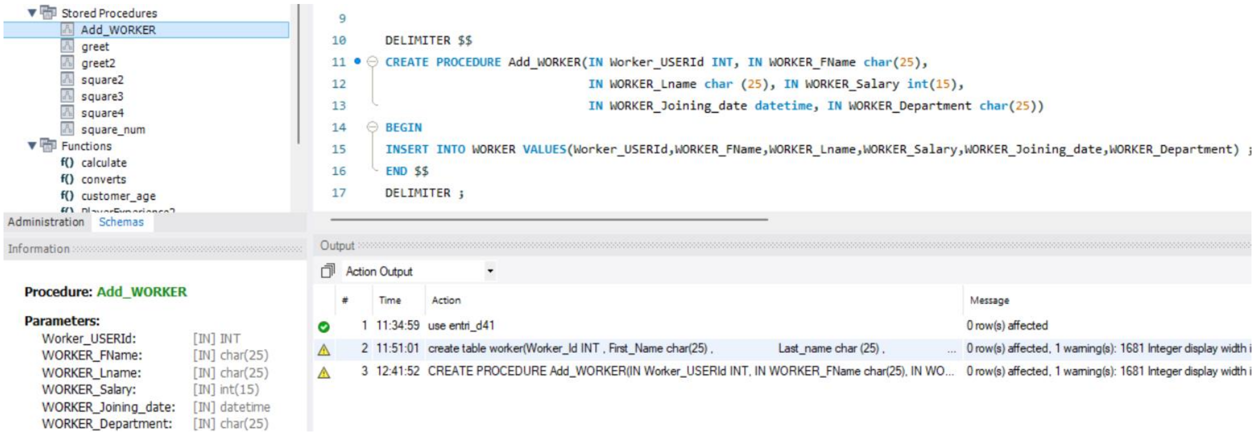
Table: worker

Columns:

Worker_Id	int
First_Name	char(25)
Last_name	char(25)
Salary	int
Joining_date	datetime
Department	char(25)

```
    #  Time      Action                                                                              Message
✔  1  11:34:59  use entri_d41                                                                       0 row(s) affected
⚠  2  11:51:01  create table worker(Worker_Id INT , First_Name char(25) ,      Last_name char (25) ,  ...  0 row(s) affected, 1 wa
```

1. Create a stored procedure that takes in IN parameters for all the columns in the Worker table and adds a new record to the table and then invokes the procedure call.



```
DELIMITER $$
CREATE PROCEDURE Add_WORKER(IN Worker_USERId INT, IN WORKER_FName char(25),
                            IN WORKER_Lname char (25), IN WORKER_Salary int(15),
                            IN WORKER_Joining_date datetime, IN WORKER_Department char(25))
BEGIN
INSERT INTO WORKER VALUES(Worker_USERId,WORKER_FName,WORKER_Lname,WORKER_Salary,WORKER_Joining_date,WORKER_Department) ;
END $$
DELIMITER ;
```

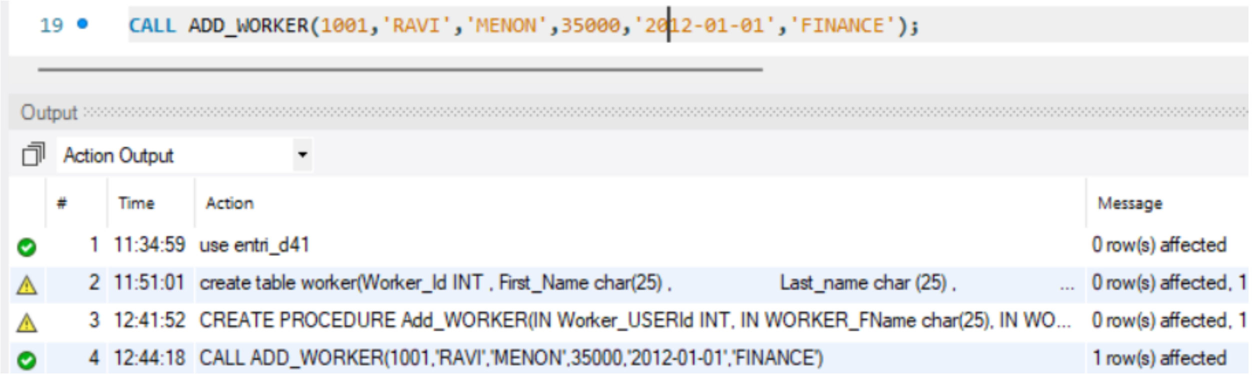
Procedure: Add_WORKER

Parameters:

Worker_USERId:	[IN] INT
WORKER_FName:	[IN] char(25)
WORKER_Lname:	[IN] char(25)
WORKER_Salary:	[IN] int(15)
WORKER_Joining_date:	[IN] datetime
WORKER_Department:	[IN] char(25)

```
    #  Time      Action                                                                          Message
✔  1  11:34:59  use entri_d41                                                                   0 row(s) affected
⚠  2  11:51:01  create table worker(Worker_Id INT , First_Name char(25) ,     Last_name char (25) ,  ...  0 row(s) affected, 1 warning(s): 1681 Integer display width i
⚠  3  12:41:52  CREATE PROCEDURE Add_WORKER(IN Worker_USERId INT, IN WORKER_FName char(25), IN WO...  0 row(s) affected, 1 warning(s): 1681 Integer display width i
```

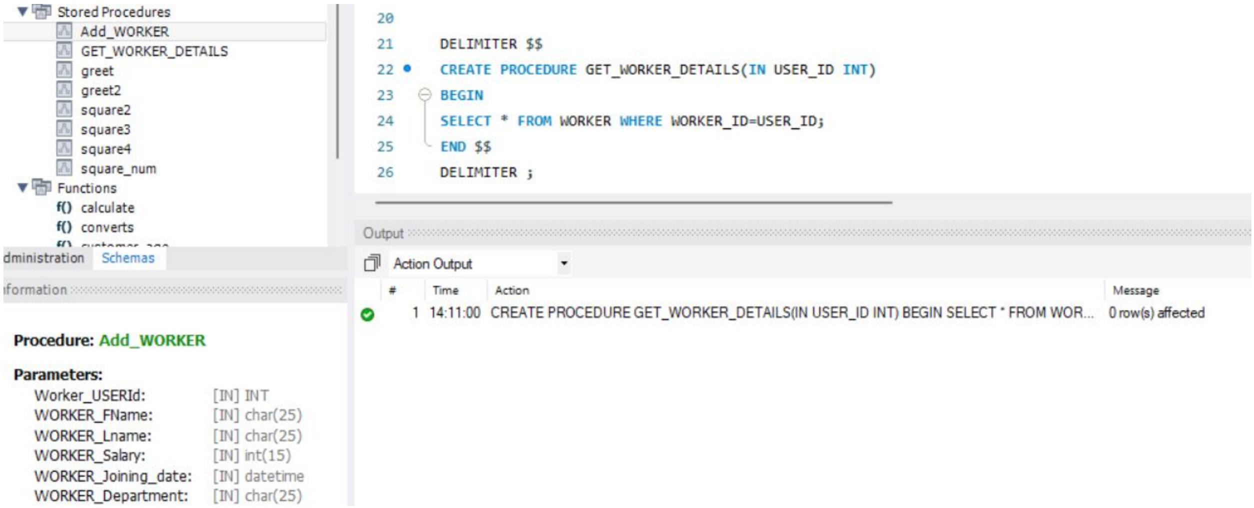
INSERT VALUES INTO THE WORKER TABLE USING THE STORE PROCEDURE



```
CALL ADD_WORKER(1001,'RAVI','MENON',35000,'2012-01-01','FINANCE');
```

```
   #  Time      Action                                                                              Message
✓  1  11:34:59  use entri_d41                                                                       0 row(s) affected
⚠  2  11:51:01  create table worker(Worker_Id INT , First_Name char(25) ,          Last_name char (25) ,   ...  0 row(s) affected, 1
⚠  3  12:41:52  CREATE PROCEDURE Add_WORKER(IN Worker_USERId INT, IN WORKER_FName char(25), IN WO...   0 row(s) affected, 1
✓  4  12:44:18  CALL ADD_WORKER(1001,'RAVI','MENON',35000,'2012-01-01','FINANCE')                   1 row(s) affected
```

CREATE PROCEDURE FOR GET DETAILS OF THE WORKER



```

DELIMITER $$
CREATE PROCEDURE GET_WORKER_DETAILS(IN USER_ID INT)
BEGIN
SELECT * FROM WORKER WHERE WORKER_ID=USER_ID;
END $$
DELIMITER ;
```

```
   #  Time      Action                                                                   Message
✓  1  14:11:00  CREATE PROCEDURE GET_WORKER_DETAILS(IN USER_ID INT) BEGIN SELECT * FROM WOR...  0 row(s) affected
```

Procedure: Add_WORKER

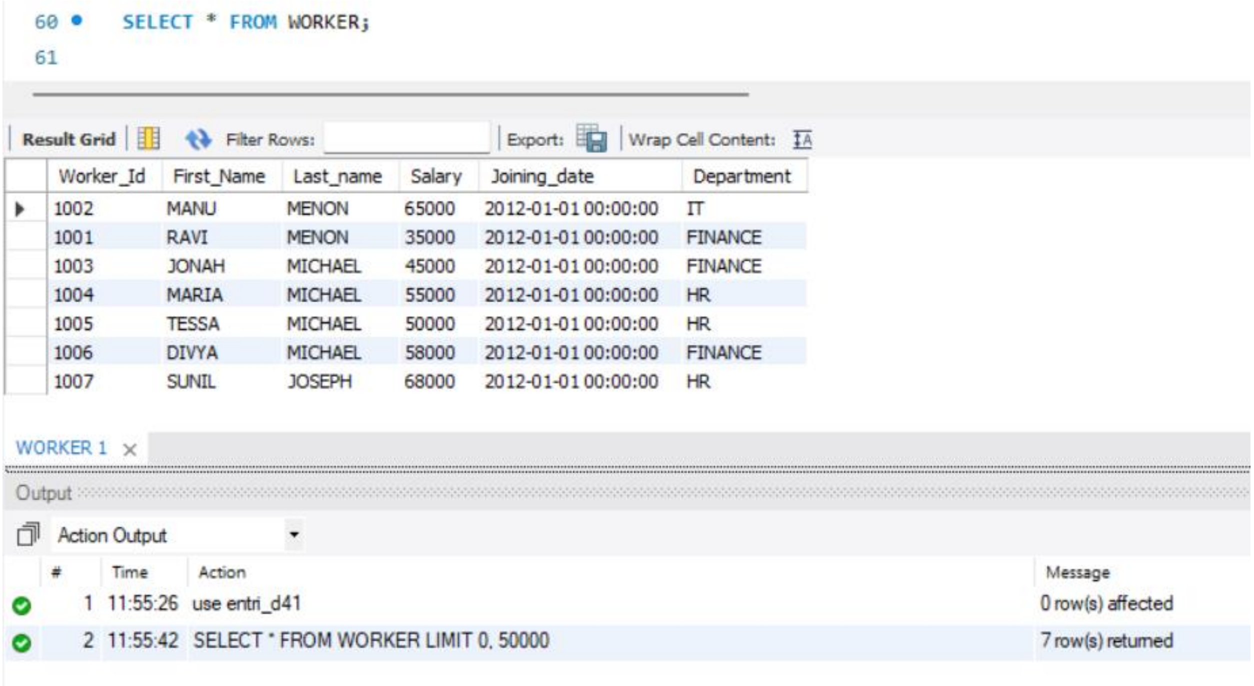
Parameters:

Worker_USERId:	[IN] INT
WORKER_FName:	[IN] char(25)
WORKER_Lname:	[IN] char(25)
WORKER_Salary:	[IN] int(15)
WORKER_Joining_date:	[IN] datetime
WORKER_Department:	[IN] char(25)

GET THE DETAILS OF THE WORKER WITH A SPECIFIC USER_ID

```
DELIMITER $$
CREATE PROCEDURE GET_WORKER_DETAILS(IN USER_ID INT)
BEGIN
SELECT * FROM WORKER WHERE WORKER_ID=USER_ID;
END $$
DELIMITER ;
```

CALL GET_WORKER_DETAILS(1002);

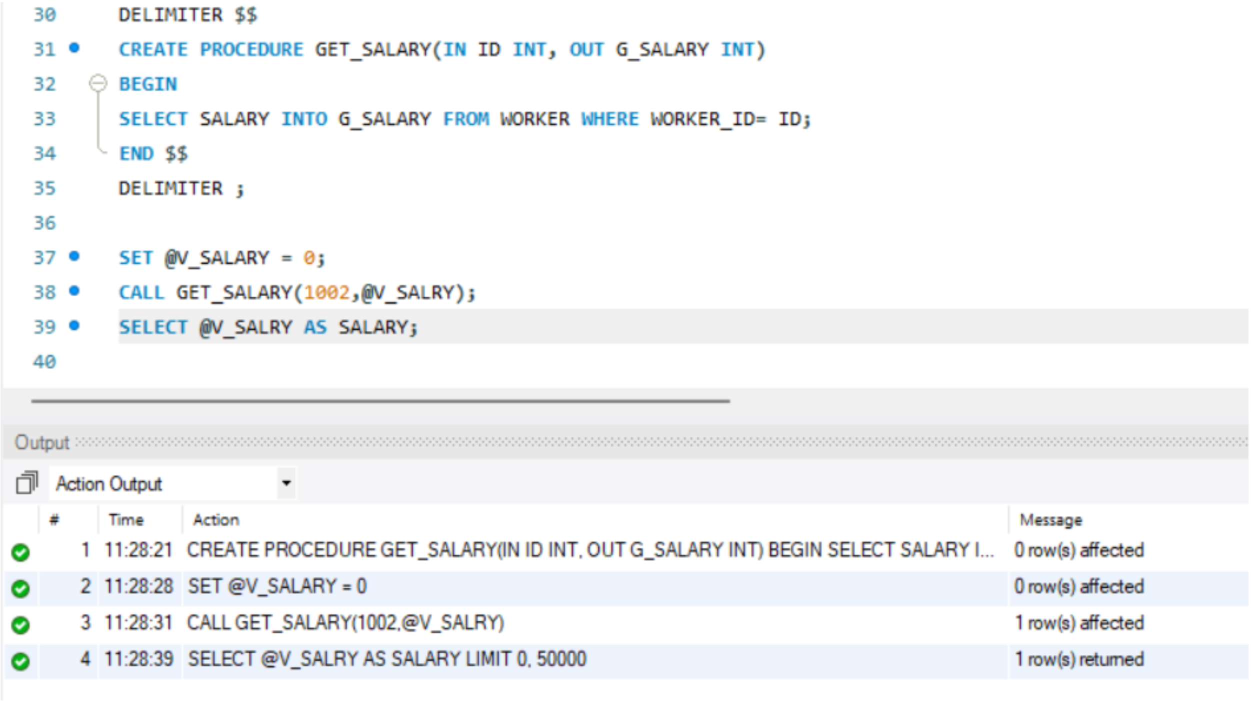


```
SELECT * FROM WORKER;
```

Worker_Id	First_Name	Last_name	Salary	Joining_date	Department
1002	MANU	MENON	65000	2012-01-01 00:00:00	IT
1001	RAVI	MENON	35000	2012-01-01 00:00:00	FINANCE
1003	JONAH	MICHAEL	45000	2012-01-01 00:00:00	FINANCE
1004	MARIA	MICHAEL	55000	2012-01-01 00:00:00	HR
1005	TESSA	MICHAEL	50000	2012-01-01 00:00:00	HR
1006	DIVYA	MICHAEL	58000	2012-01-01 00:00:00	FINANCE
1007	SUNIL	JOSEPH	68000	2012-01-01 00:00:00	HR

#	Time	Action	Message
1	11:55:26	use entri_d41	0 row(s) affected
2	11:55:42	SELECT * FROM WORKER LIMIT 0, 50000	7 row(s) returned

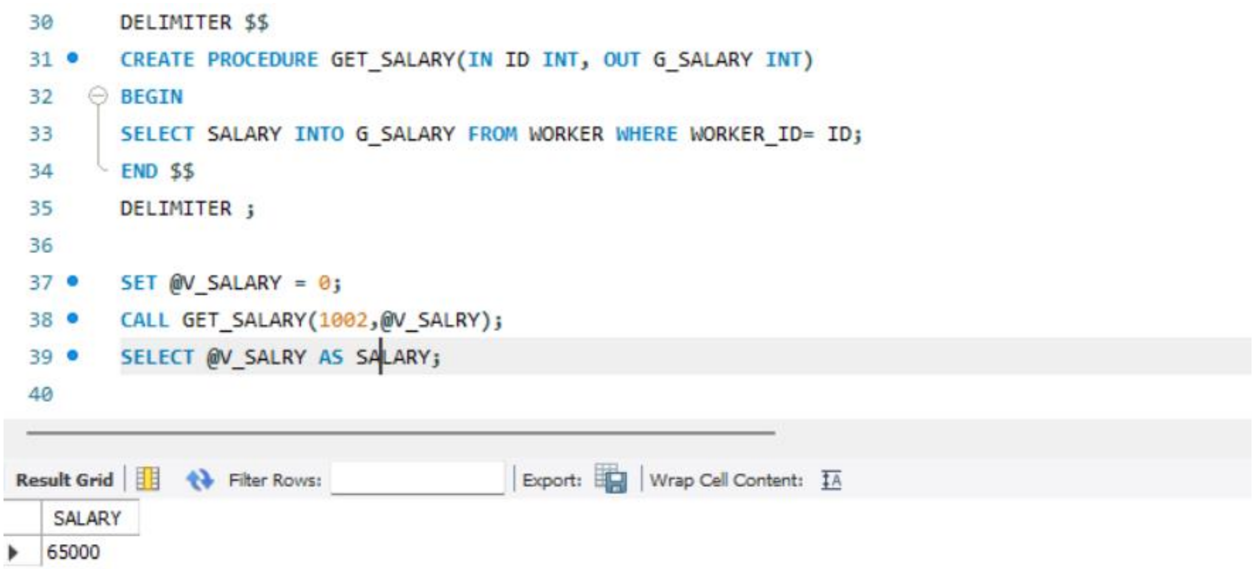
2. Write stored procedure takes in an IN parameter for WORKER_ID and an OUT parameter for SALARY. It should retrieve the salary of the worker with the given ID and returns it in the p_salary parameter. Then make the procedure call.



```
DELIMITER $$
CREATE PROCEDURE GET_SALARY(IN ID INT, OUT G_SALARY INT)
BEGIN
SELECT SALARY INTO G_SALARY FROM WORKER WHERE WORKER_ID= ID;
END $$
DELIMITER ;

SET @V_SALARY = 0;
CALL GET_SALARY(1002,@V_SALRY);
SELECT @V_SALRY AS SALARY;
```

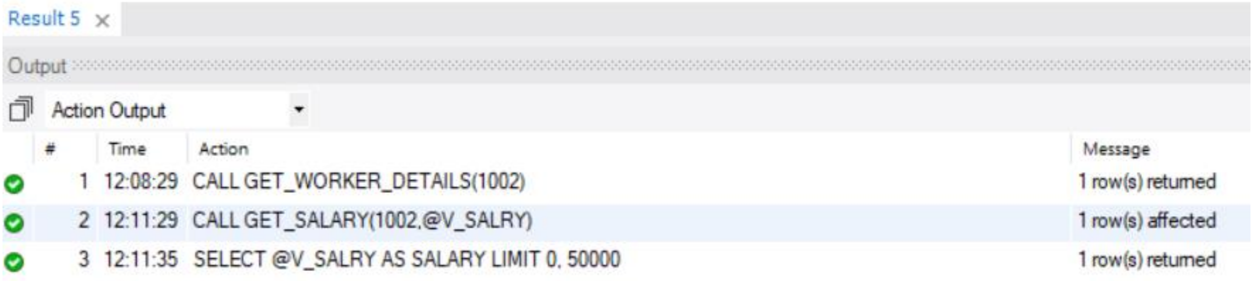
	#	Time	Action	Message
✓	1	11:28:21	CREATE PROCEDURE GET_SALARY(IN ID INT, OUT G_SALARY INT) BEGIN SELECT SALARY I...	0 row(s) affected
✓	2	11:28:28	SET @V_SALARY = 0	0 row(s) affected
✓	3	11:28:31	CALL GET_SALARY(1002,@V_SALRY)	1 row(s) affected
✓	4	11:28:39	SELECT @V_SALRY AS SALARY LIMIT 0, 50000	1 row(s) returned



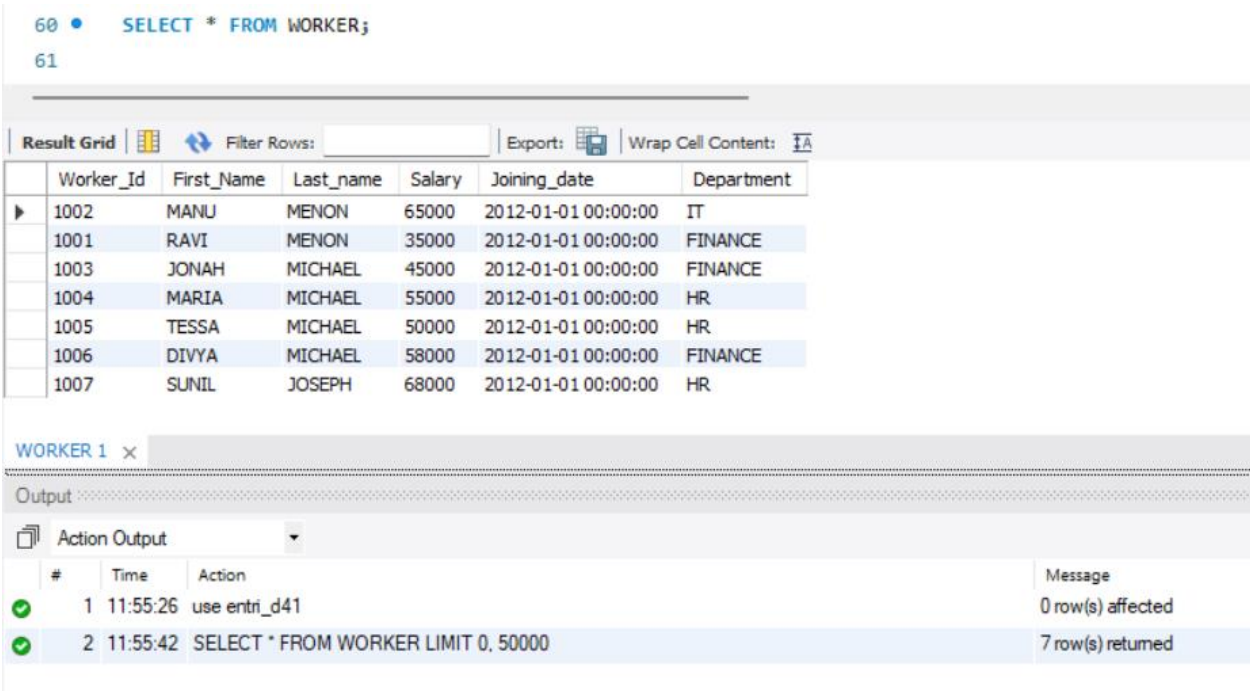
```
DELIMITER $$
CREATE PROCEDURE GET_SALARY(IN ID INT, OUT G_SALARY INT)
BEGIN
SELECT SALARY INTO G_SALARY FROM WORKER WHERE WORKER_ID= ID;
END $$
DELIMITER ;

SET @V_SALARY = 0;
CALL GET_SALARY(1002,@V_SALRY);
SELECT @V_SALRY AS SALARY;
```

SALARY
65000



#	Time	Action	Message
1	12:08:29	CALL GET_WORKER_DETAILS(1002)	1 row(s) returned
2	12:11:29	CALL GET_SALARY(1002,@V_SALRY)	1 row(s) affected
3	12:11:35	SELECT @V_SALRY AS SALARY LIMIT 0, 50000	1 row(s) returned

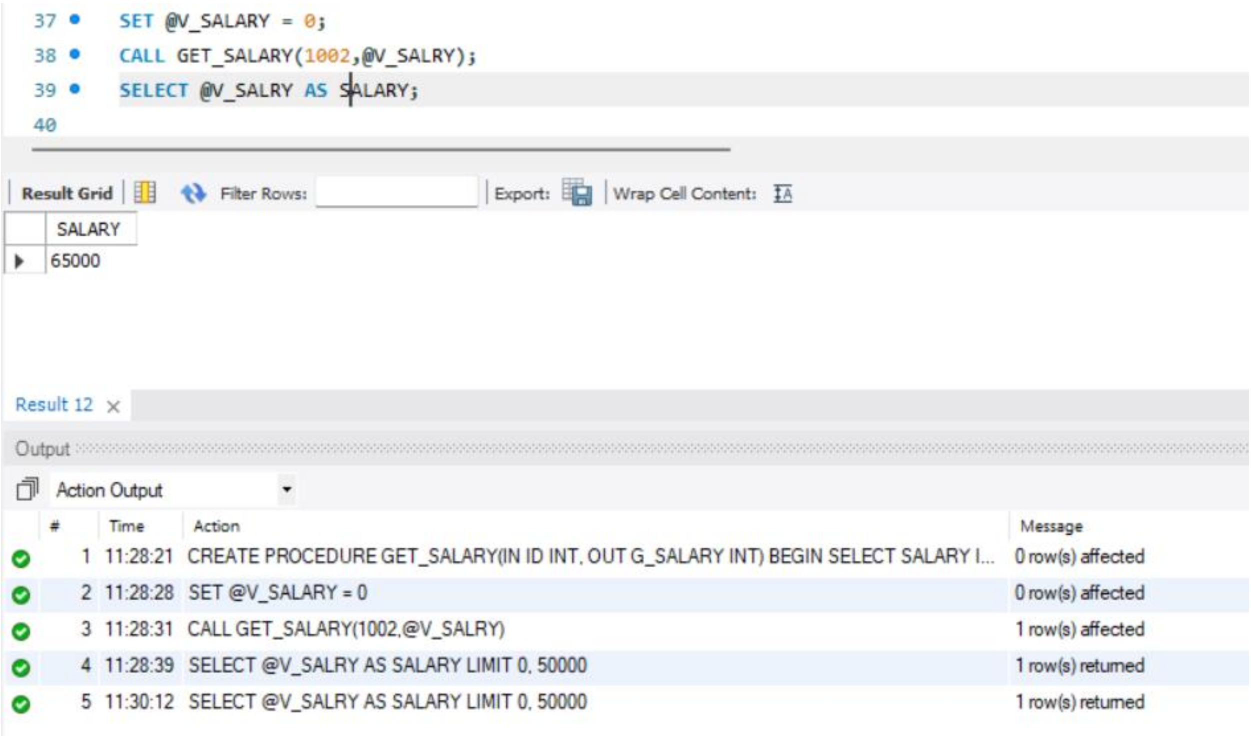


```
SELECT * FROM WORKER;
```

Worker_Id	First_Name	Last_name	Salary	Joining_date	Department
1002	MANU	MENON	65000	2012-01-01 00:00:00	IT
1001	RAVI	MENON	35000	2012-01-01 00:00:00	FINANCE
1003	JONAH	MICHAEL	45000	2012-01-01 00:00:00	FINANCE
1004	MARIA	MICHAEL	55000	2012-01-01 00:00:00	HR
1005	TESSA	MICHAEL	50000	2012-01-01 00:00:00	HR
1006	DIVYA	MICHAEL	58000	2012-01-01 00:00:00	FINANCE
1007	SUNIL	JOSEPH	68000	2012-01-01 00:00:00	HR

#	Time	Action	Message
1	11:55:26	use entri_d41	0 row(s) affected
2	11:55:42	SELECT * FROM WORKER LIMIT 0, 50000	7 row(s) returned

CALL procedure to retrieve the salary of the worker with the given ID

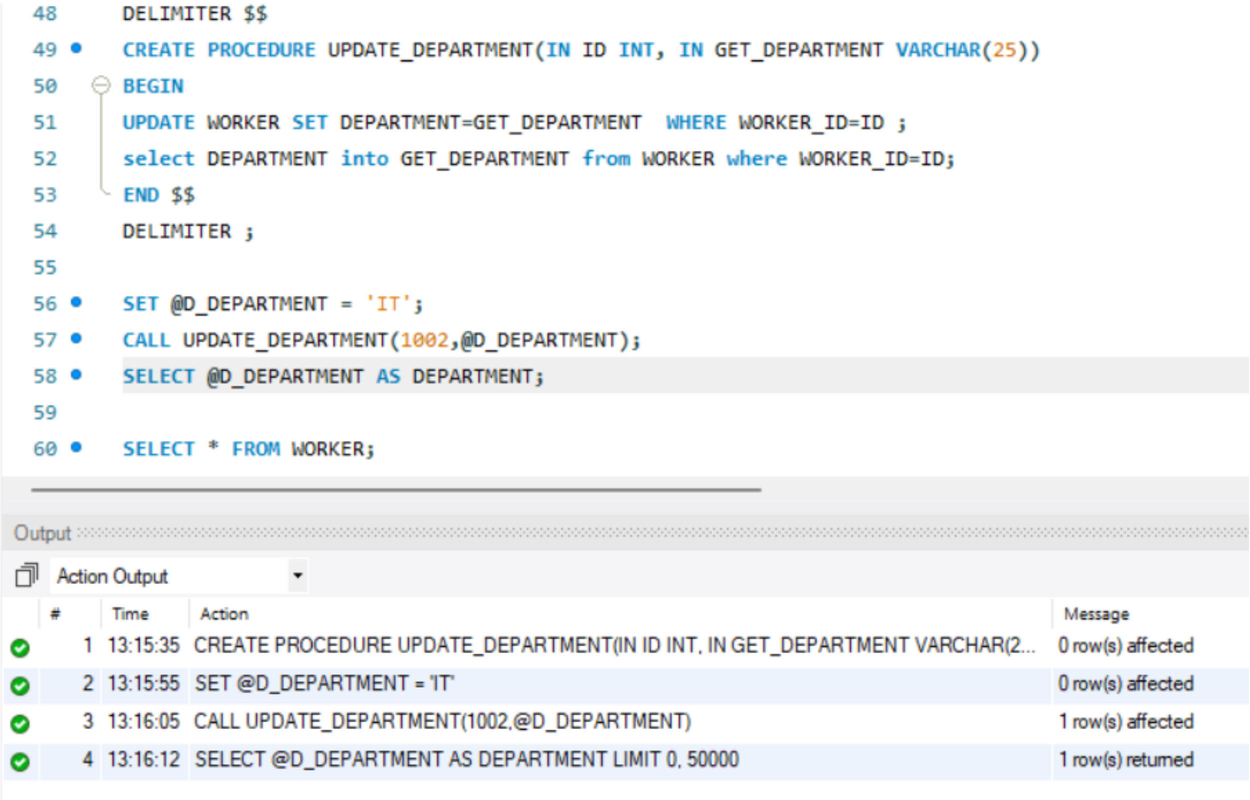


```
SET @V_SALARY = 0;
CALL GET_SALARY(1002,@V_SALRY);
SELECT @V_SALRY AS SALARY;
```

SALARY
65000

#	Time	Action	Message
1	11:28:21	CREATE PROCEDURE GET_SALARY(IN ID INT, OUT G_SALARY INT) BEGIN SELECT SALARY I...	0 row(s) affected
2	11:28:28	SET @V_SALARY = 0	0 row(s) affected
3	11:28:31	CALL GET_SALARY(1002,@V_SALRY)	1 row(s) affected
4	11:28:39	SELECT @V_SALRY AS SALARY LIMIT 0, 50000	1 row(s) returned
5	11:30:12	SELECT @V_SALRY AS SALARY LIMIT 0, 50000	1 row(s) returned

3. Create a stored procedure that takes in IN parameters for WORKER_ID and DEPARTMENT. It should update the department of the worker with the given ID. Then make a procedure call.



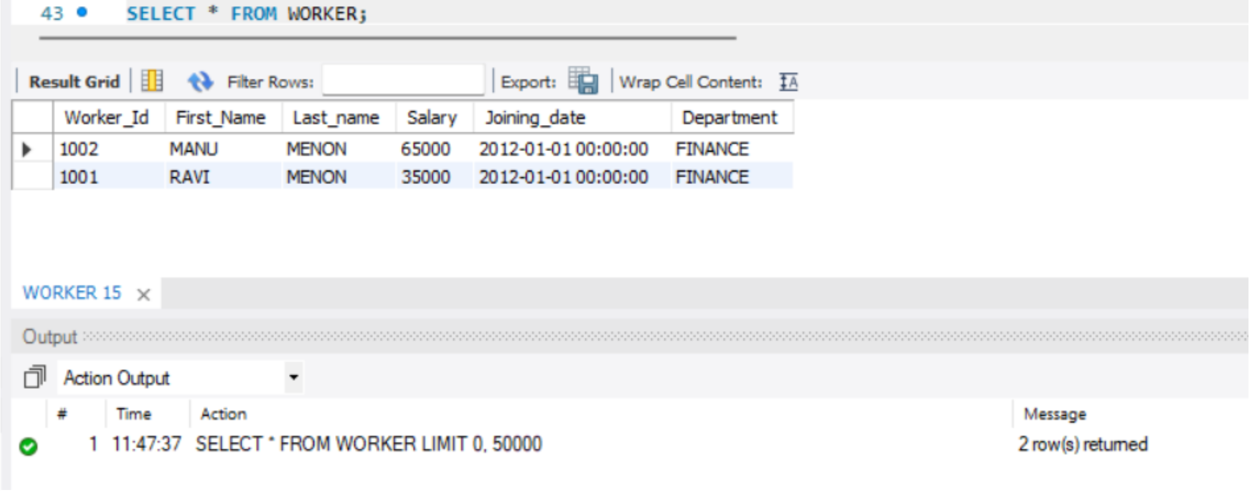
```
DELIMITER $$
CREATE PROCEDURE UPDATE_DEPARTMENT(IN ID INT, IN GET_DEPARTMENT VARCHAR(25))
BEGIN
UPDATE WORKER SET DEPARTMENT=GET_DEPARTMENT  WHERE WORKER_ID=ID ;
select DEPARTMENT into GET_DEPARTMENT from WORKER where WORKER_ID=ID;
END $$
DELIMITER ;

SET @D_DEPARTMENT = 'IT';
CALL UPDATE_DEPARTMENT(1002,@D_DEPARTMENT);
SELECT @D_DEPARTMENT AS DEPARTMENT;

SELECT * FROM WORKER;
```

```
#   Time      Action                                                                      Message
1   13:15:35  CREATE PROCEDURE UPDATE_DEPARTMENT(IN ID INT, IN GET_DEPARTMENT VARCHAR(2...   0 row(s) affected
2   13:15:55  SET @D_DEPARTMENT = 'IT'                                                    0 row(s) affected
3   13:16:05  CALL UPDATE_DEPARTMENT(1002,@D_DEPARTMENT)                                  1 row(s) affected
4   13:16:12  SELECT @D_DEPARTMENT AS DEPARTMENT LIMIT 0, 50000                           1 row(s) returned
```

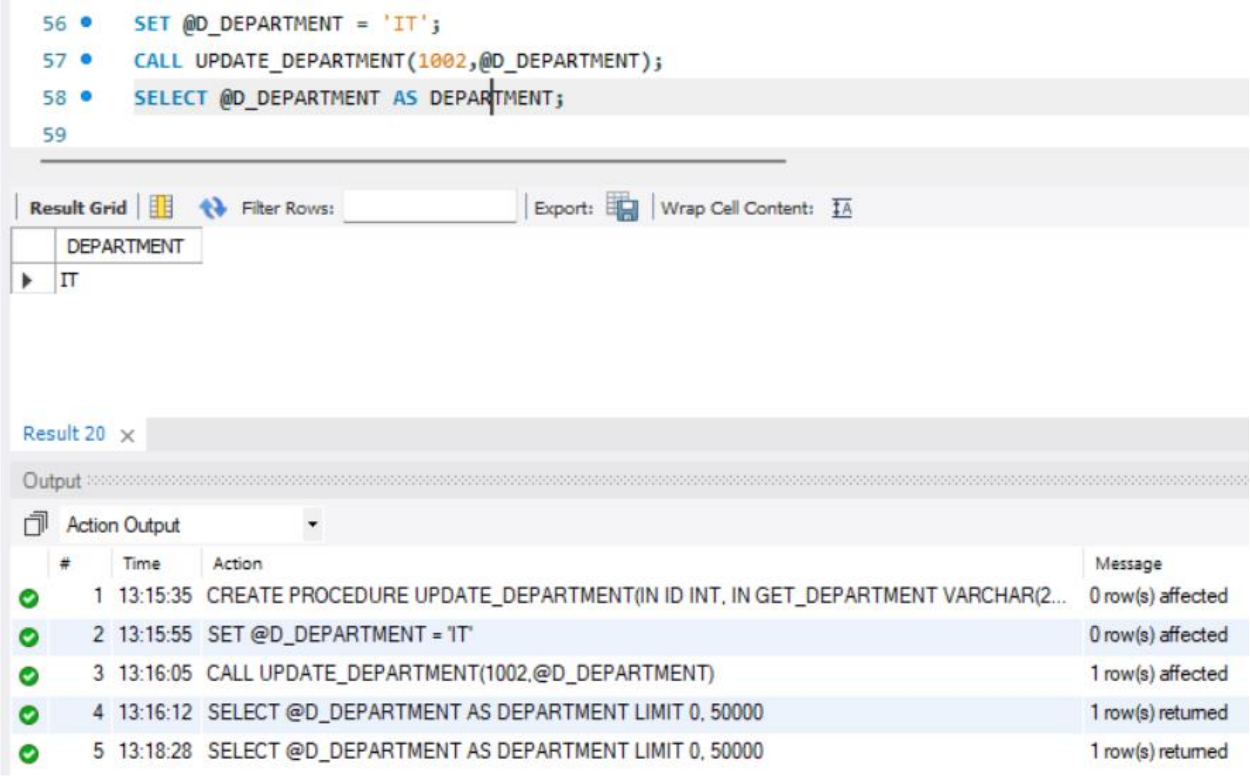
The department of the worer_id (1002) before updating was 'FINANCE' AS shown below



```
SELECT * FROM WORKER;
```

Worker_Id	First_Name	Last_name	Salary	Joining_date	Department
1002	MANU	MENON	65000	2012-01-01 00:00:00	FINANCE
1001	RAVI	MENON	35000	2012-01-01 00:00:00	FINANCE

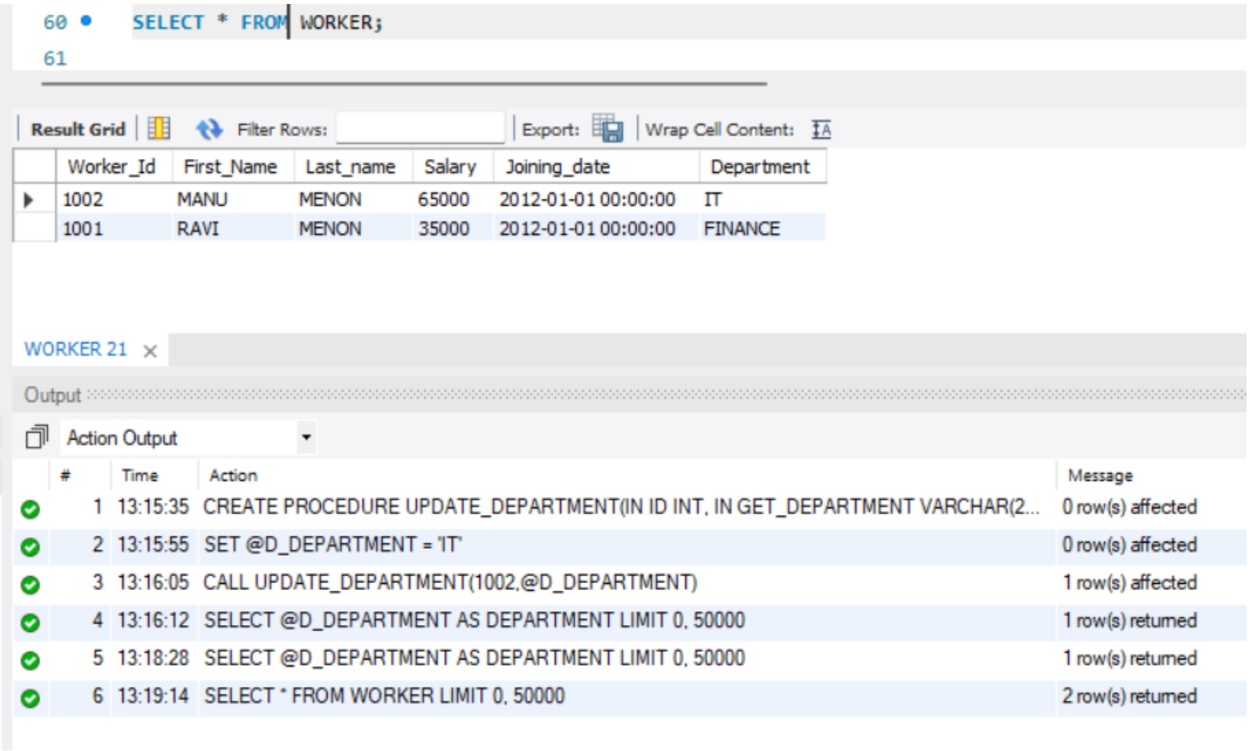
```
#   Time      Action                              Message
1   11:47:37  SELECT * FROM WORKER LIMIT 0, 50000  2 row(s) returned
```



```sql
SET @D_DEPARTMENT = 'IT';
CALL UPDATE_DEPARTMENT(1002,@D_DEPARTMENT);
SELECT @D_DEPARTMENT AS DEPARTMENT;
```

DEPARTMENT
IT

	#	Time	Action	Message
✓	1	13:15:35	CREATE PROCEDURE UPDATE_DEPARTMENT(IN ID INT, IN GET_DEPARTMENT VARCHAR(2...	0 row(s) affected
✓	2	13:15:55	SET @D_DEPARTMENT = 'IT'	0 row(s) affected
✓	3	13:16:05	CALL UPDATE_DEPARTMENT(1002,@D_DEPARTMENT)	1 row(s) affected
✓	4	13:16:12	SELECT @D_DEPARTMENT AS DEPARTMENT LIMIT 0, 50000	1 row(s) returned
✓	5	13:18:28	SELECT @D_DEPARTMENT AS DEPARTMENT LIMIT 0, 50000	1 row(s) returned

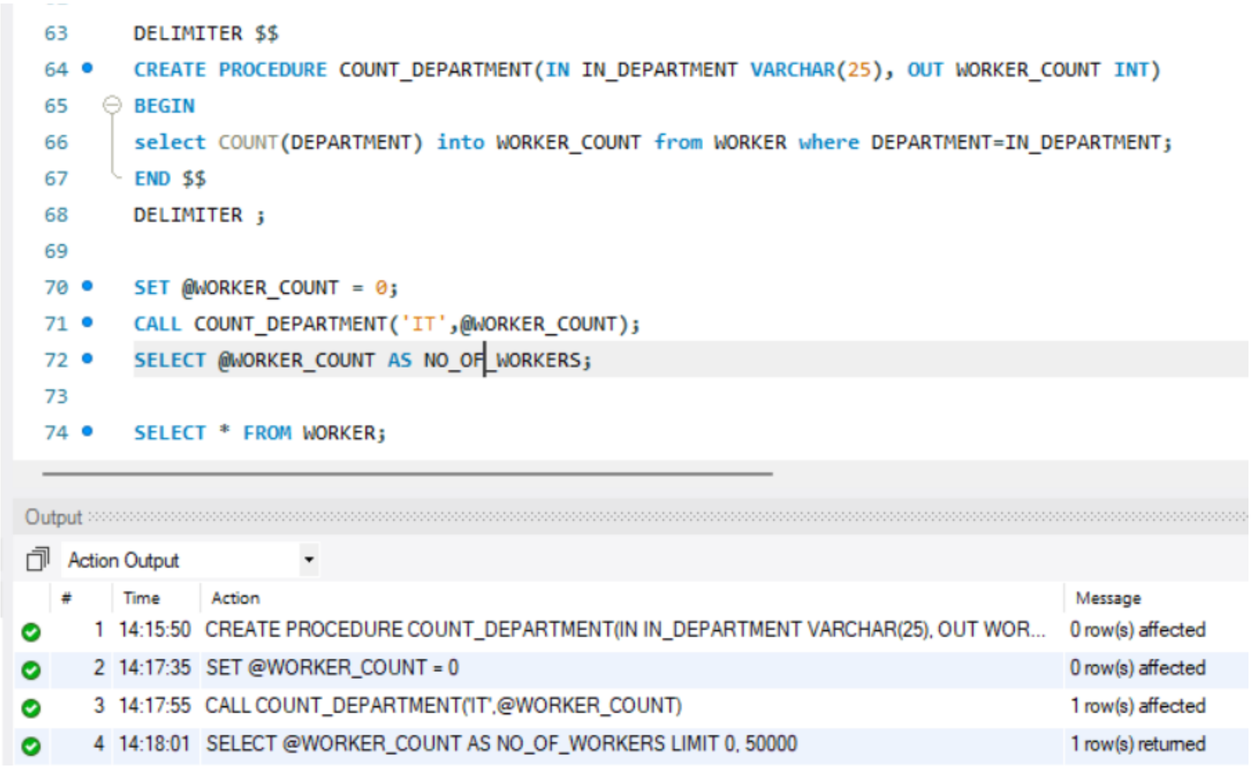


```sql
SELECT * FROM WORKER;
```

Worker_Id	First_Name	Last_name	Salary	Joining_date	Department
1002	MANU	MENON	65000	2012-01-01 00:00:00	IT
1001	RAVI	MENON	35000	2012-01-01 00:00:00	FINANCE

	#	Time	Action	Message
✓	1	13:15:35	CREATE PROCEDURE UPDATE_DEPARTMENT(IN ID INT, IN GET_DEPARTMENT VARCHAR(2...	0 row(s) affected
✓	2	13:15:55	SET @D_DEPARTMENT = 'IT'	0 row(s) affected
✓	3	13:16:05	CALL UPDATE_DEPARTMENT(1002,@D_DEPARTMENT)	1 row(s) affected
✓	4	13:16:12	SELECT @D_DEPARTMENT AS DEPARTMENT LIMIT 0, 50000	1 row(s) returned
✓	5	13:18:28	SELECT @D_DEPARTMENT AS DEPARTMENT LIMIT 0, 50000	1 row(s) returned
✓	6	13:19:14	SELECT * FROM WORKER LIMIT 0, 50000	2 row(s) returned

4. Write a stored procedure that takes in an IN parameter for DEPARTMENT and an OUT parameter for p_workerCount. It should retrieve the number of workers in the given department and returns it in the p_workerCount parameter. Make procedure call.

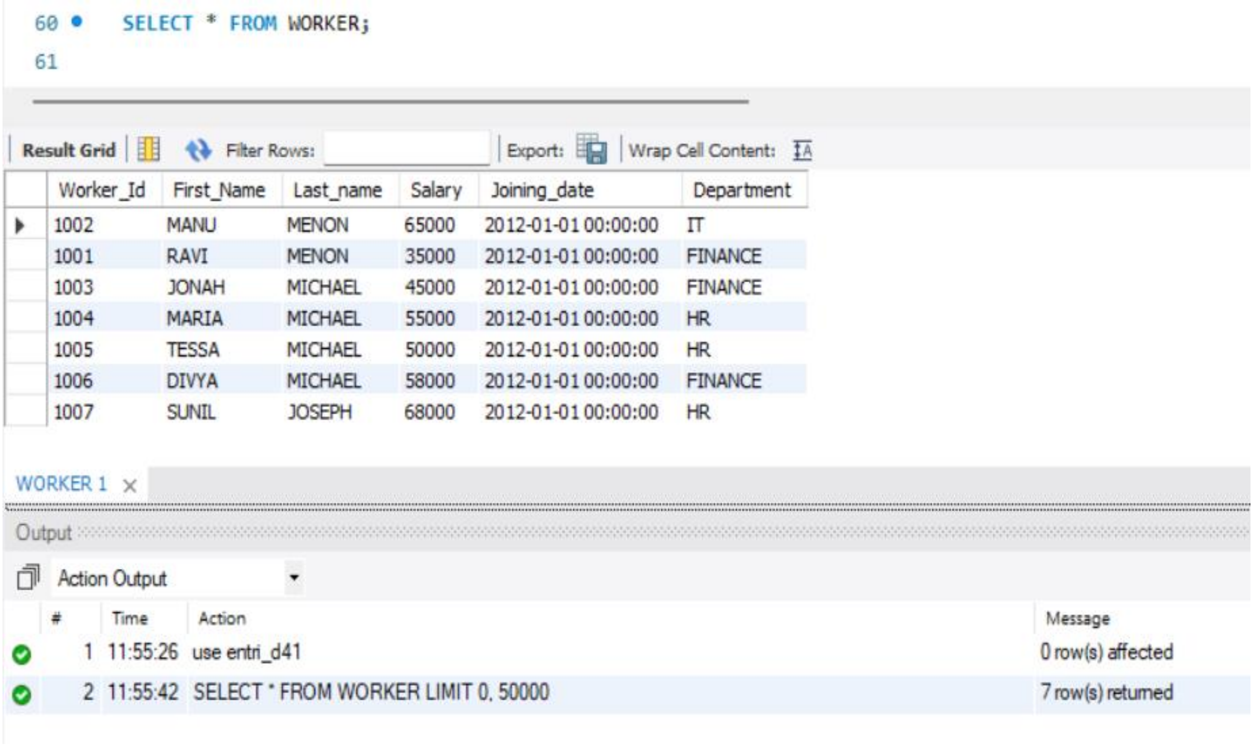


```
DELIMITER $$
CREATE PROCEDURE COUNT_DEPARTMENT(IN IN_DEPARTMENT VARCHAR(25), OUT WORKER_COUNT INT)
BEGIN
select COUNT(DEPARTMENT) into WORKER_COUNT from WORKER where DEPARTMENT=IN_DEPARTMENT;
END $$
DELIMITER ;

SET @WORKER_COUNT = 0;
CALL COUNT_DEPARTMENT('IT',@WORKER_COUNT);
SELECT @WORKER_COUNT AS NO_OF_WORKERS;

SELECT * FROM WORKER;
```

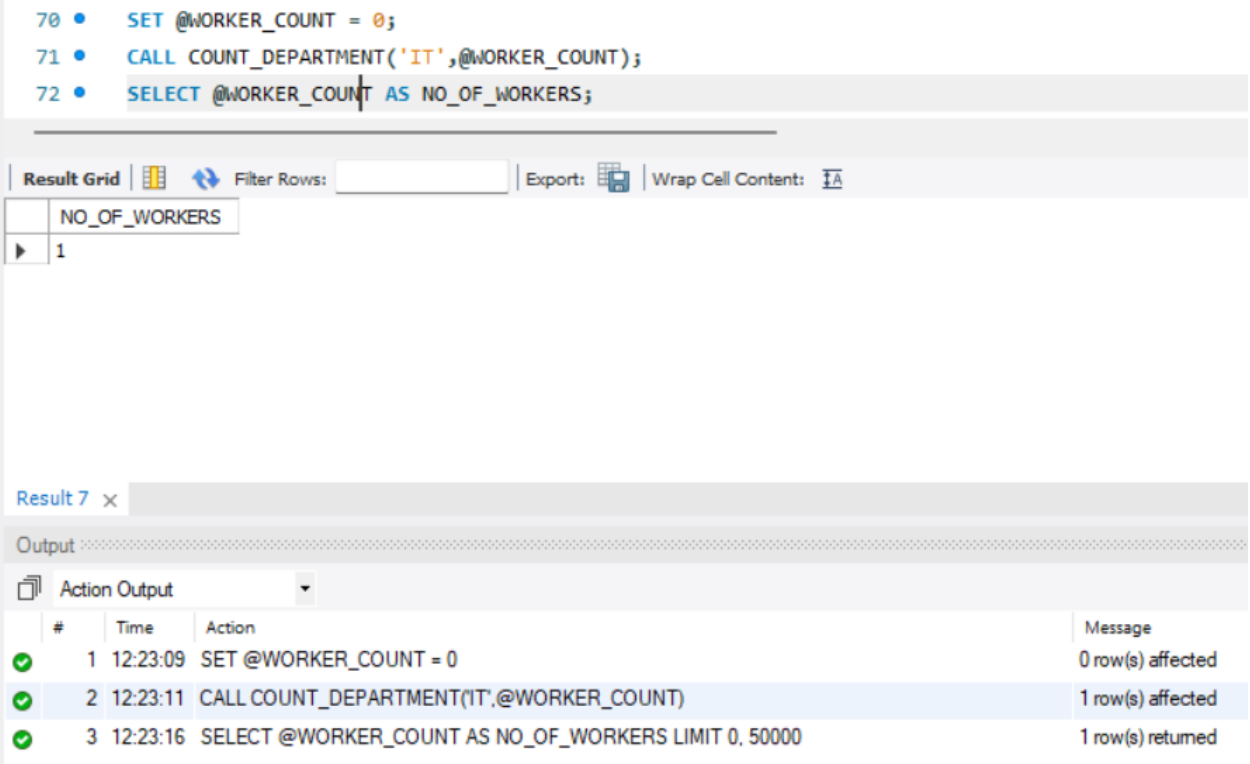
	#	Time	Action	Message
✓	1	14:15:50	CREATE PROCEDURE COUNT_DEPARTMENT(IN IN_DEPARTMENT VARCHAR(25), OUT WOR...	0 row(s) affected
✓	2	14:17:35	SET @WORKER_COUNT = 0	0 row(s) affected
✓	3	14:17:55	CALL COUNT_DEPARTMENT('IT',@WORKER_COUNT)	1 row(s) affected
✓	4	14:18:01	SELECT @WORKER_COUNT AS NO_OF_WORKERS LIMIT 0, 50000	1 row(s) returned



```
SELECT * FROM WORKER;
```

Worker_Id	First_Name	Last_name	Salary	Joining_date	Department
1002	MANU	MENON	65000	2012-01-01 00:00:00	IT
1001	RAVI	MENON	35000	2012-01-01 00:00:00	FINANCE
1003	JONAH	MICHAEL	45000	2012-01-01 00:00:00	FINANCE
1004	MARIA	MICHAEL	55000	2012-01-01 00:00:00	HR
1005	TESSA	MICHAEL	50000	2012-01-01 00:00:00	HR
1006	DIVYA	MICHAEL	58000	2012-01-01 00:00:00	FINANCE
1007	SUNIL	JOSEPH	68000	2012-01-01 00:00:00	HR

#	Time	Action	Message
1	11:55:26	use entri_d41	0 row(s) affected
2	11:55:42	SELECT * FROM WORKER LIMIT 0, 50000	7 row(s) returned

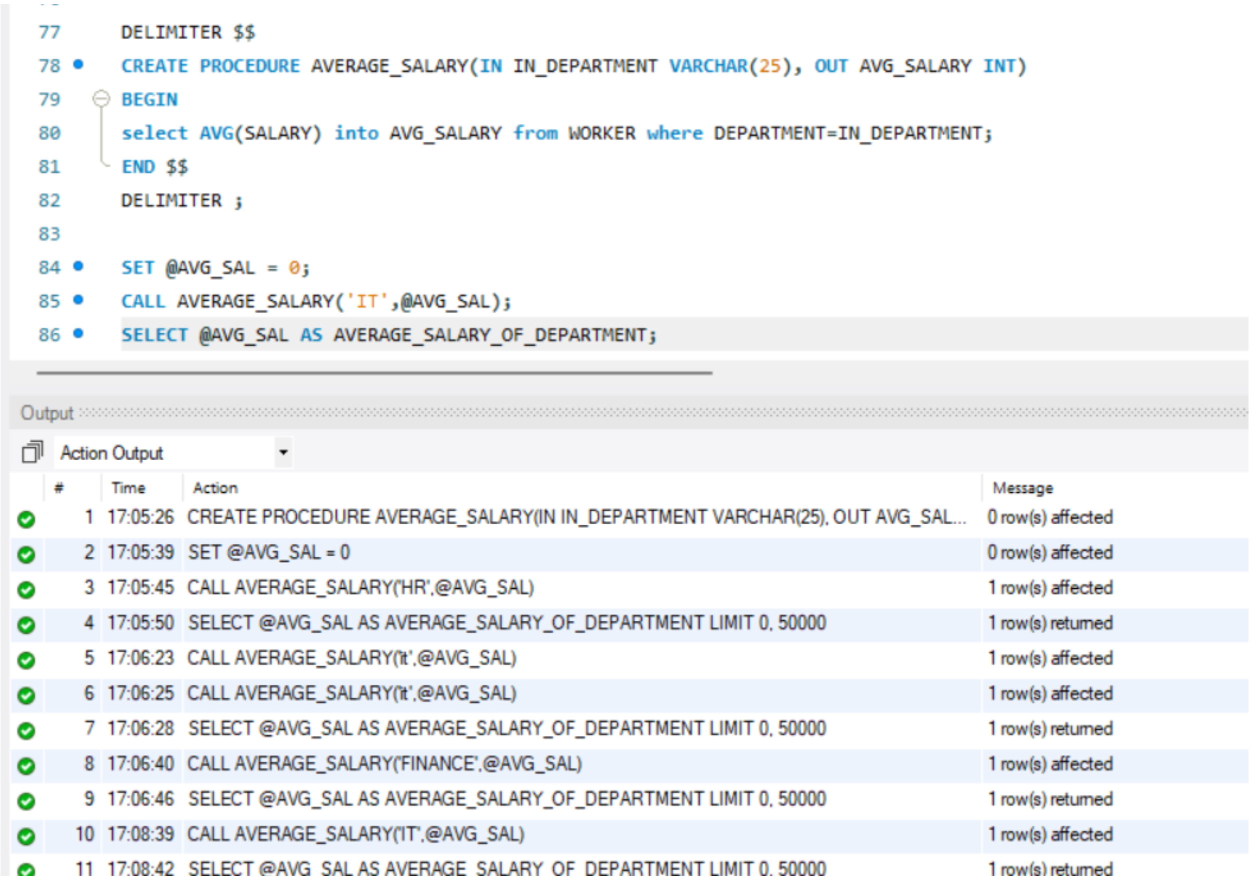


```
SET @WORKER_COUNT = 0;
CALL COUNT_DEPARTMENT('IT',@WORKER_COUNT);
SELECT @WORKER_COUNT AS NO_OF_WORKERS;
```

NO_OF_WORKERS
1

#	Time	Action	Message
1	12:23:09	SET @WORKER_COUNT = 0	0 row(s) affected
2	12:23:11	CALL COUNT_DEPARTMENT('IT',@WORKER_COUNT)	1 row(s) affected
3	12:23:16	SELECT @WORKER_COUNT AS NO_OF_WORKERS LIMIT 0, 50000	1 row(s) returned

5. Write a stored procedure that takes in an IN parameter for DEPARTMENT and an OUT parameter for p_avgSalary. It should retrieve the average salary of all workers in the given department and returns it in the p_avgSalary parameter and call the procedure.

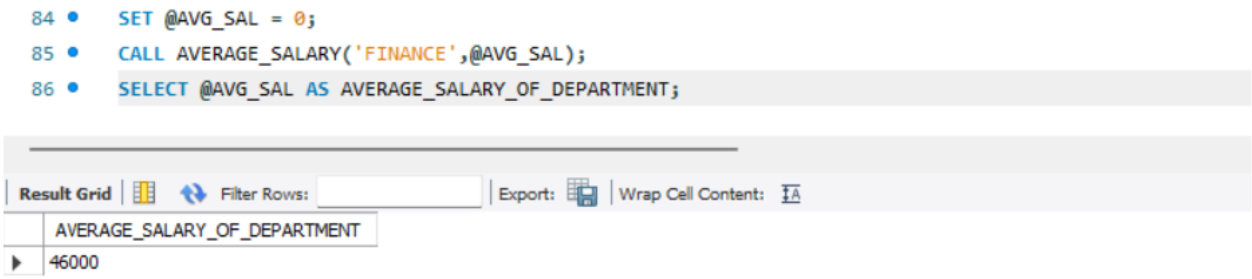


```
DELIMITER $$
CREATE PROCEDURE AVERAGE_SALARY(IN IN_DEPARTMENT VARCHAR(25), OUT AVG_SALARY INT)
BEGIN
select AVG(SALARY) into AVG_SALARY from WORKER where DEPARTMENT=IN_DEPARTMENT;
END $$
DELIMITER ;

SET @AVG_SAL = 0;
CALL AVERAGE_SALARY('IT',@AVG_SAL);
SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT;
```

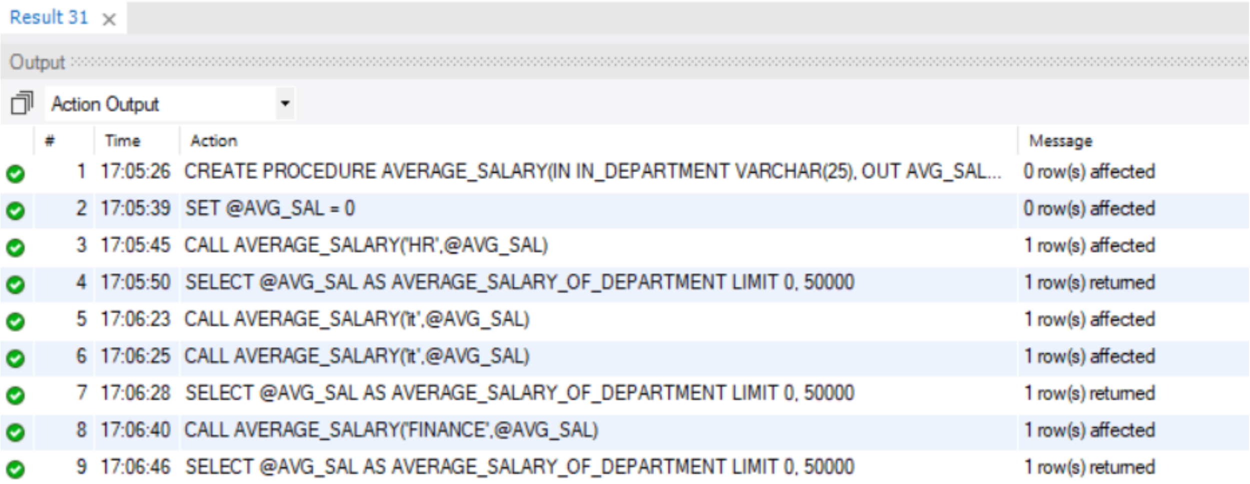
```
 #   Time      Action                                                                         Message
 1   17:05:26  CREATE PROCEDURE AVERAGE_SALARY(IN IN_DEPARTMENT VARCHAR(25), OUT AVG_SAL...   0 row(s) affected
 2   17:05:39  SET @AVG_SAL = 0                                                               0 row(s) affected
 3   17:05:45  CALL AVERAGE_SALARY('HR',@AVG_SAL)                                             1 row(s) affected
 4   17:05:50  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000                 1 row(s) returned
 5   17:06:23  CALL AVERAGE_SALARY('it',@AVG_SAL)                                             1 row(s) affected
 6   17:06:25  CALL AVERAGE_SALARY('it',@AVG_SAL)                                             1 row(s) affected
 7   17:06:28  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000                 1 row(s) returned
 8   17:06:40  CALL AVERAGE_SALARY('FINANCE',@AVG_SAL)                                        1 row(s) affected
 9   17:06:46  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000                 1 row(s) returned
10   17:08:39  CALL AVERAGE_SALARY('IT',@AVG_SAL)                                             1 row(s) affected
11   17:08:42  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000                 1 row(s) returned
```

Average salary when opted the department as ‘finance’



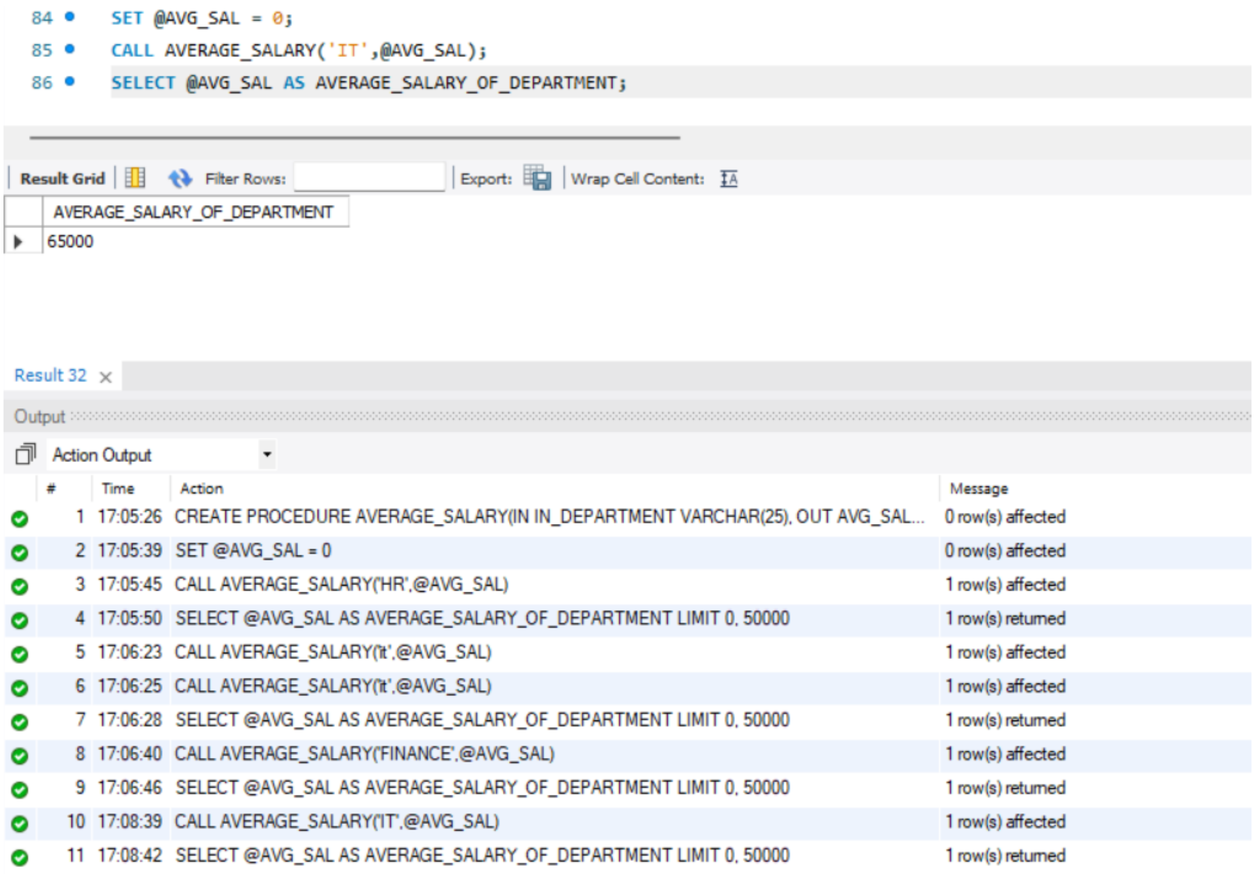
```
SET @AVG_SAL = 0;
CALL AVERAGE_SALARY('FINANCE',@AVG_SAL);
SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT;
```

AVERAGE_SALARY_OF_DEPARTMENT
46000



```
#  Time      Action                                                                   Message
1  17:05:26  CREATE PROCEDURE AVERAGE_SALARY(IN IN_DEPARTMENT VARCHAR(25), OUT AVG_SAL...  0 row(s) affected
2  17:05:39  SET @AVG_SAL = 0                                                         0 row(s) affected
3  17:05:45  CALL AVERAGE_SALARY('HR',@AVG_SAL)                                       1 row(s) affected
4  17:05:50  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000           1 row(s) returned
5  17:06:23  CALL AVERAGE_SALARY('it',@AVG_SAL)                                       1 row(s) affected
6  17:06:25  CALL AVERAGE_SALARY('it',@AVG_SAL)                                       1 row(s) affected
7  17:06:28  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000           1 row(s) returned
8  17:06:40  CALL AVERAGE_SALARY('FINANCE',@AVG_SAL)                                  1 row(s) affected
9  17:06:46  SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000           1 row(s) returned
```

Average salary when opted the department as ‘IT’

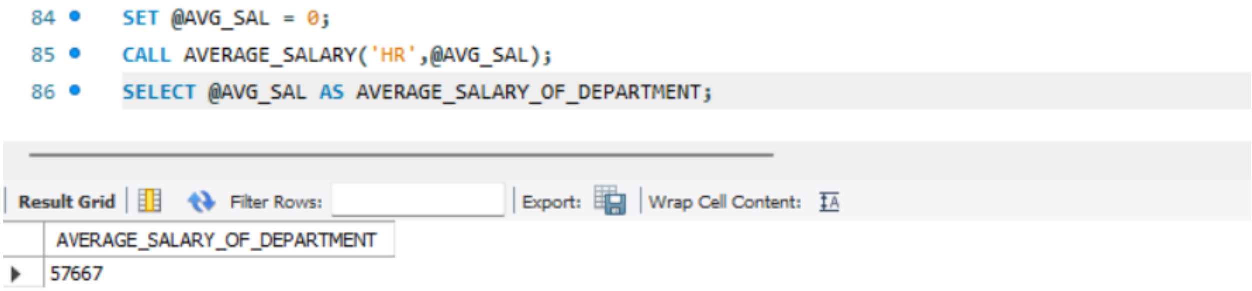


```sql
SET @AVG_SAL = 0;
CALL AVERAGE_SALARY('IT',@AVG_SAL);
SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT;
```

AVERAGE_SALARY_OF_DEPARTMENT
65000

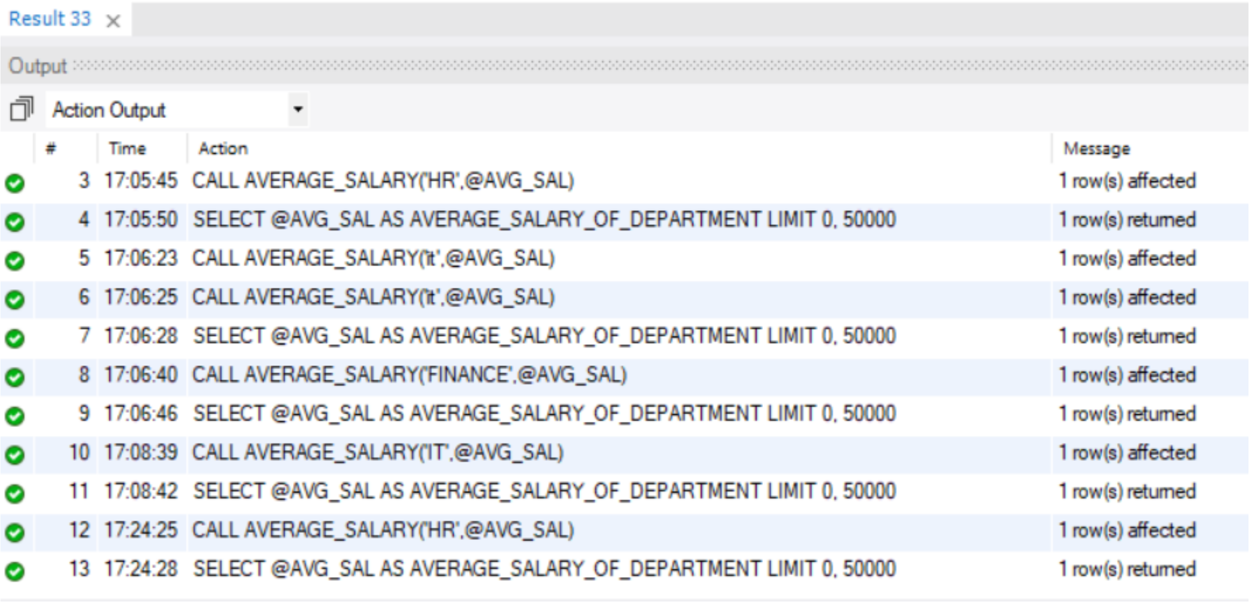
#	Time	Action	Message
1	17:05:26	CREATE PROCEDURE AVERAGE_SALARY(IN IN_DEPARTMENT VARCHAR(25), OUT AVG_SAL...	0 row(s) affected
2	17:05:39	SET @AVG_SAL = 0	0 row(s) affected
3	17:05:45	CALL AVERAGE_SALARY('HR',@AVG_SAL)	1 row(s) affected
4	17:05:50	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
5	17:06:23	CALL AVERAGE_SALARY('it',@AVG_SAL)	1 row(s) affected
6	17:06:25	CALL AVERAGE_SALARY('it',@AVG_SAL)	1 row(s) affected
7	17:06:28	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
8	17:06:40	CALL AVERAGE_SALARY('FINANCE',@AVG_SAL)	1 row(s) affected
9	17:06:46	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
10	17:08:39	CALL AVERAGE_SALARY('IT',@AVG_SAL)	1 row(s) affected
11	17:08:42	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned

Average salary when opted the department as ‘HR’



```
SET @AVG_SAL = 0;
CALL AVERAGE_SALARY('HR',@AVG_SAL);
SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT;
```

AVERAGE_SALARY_OF_DEPARTMENT
57667



#	Time	Action	Message
3	17:05:45	CALL AVERAGE_SALARY('HR',@AVG_SAL)	1 row(s) affected
4	17:05:50	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
5	17:06:23	CALL AVERAGE_SALARY('it',@AVG_SAL)	1 row(s) affected
6	17:06:25	CALL AVERAGE_SALARY('it',@AVG_SAL)	1 row(s) affected
7	17:06:28	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
8	17:06:40	CALL AVERAGE_SALARY('FINANCE',@AVG_SAL)	1 row(s) affected
9	17:06:46	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
10	17:08:39	CALL AVERAGE_SALARY('IT',@AVG_SAL)	1 row(s) affected
11	17:08:42	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned
12	17:24:25	CALL AVERAGE_SALARY('HR',@AVG_SAL)	1 row(s) affected
13	17:24:28	SELECT @AVG_SAL AS AVERAGE_SALARY_OF_DEPARTMENT LIMIT 0, 50000	1 row(s) returned