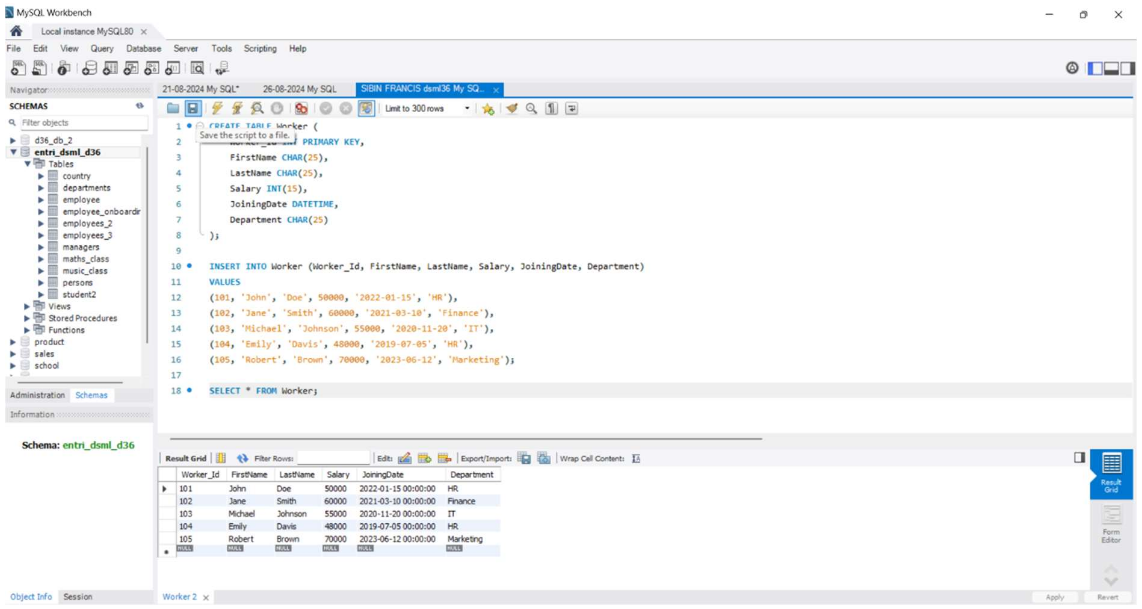
STORED PROCEDURES

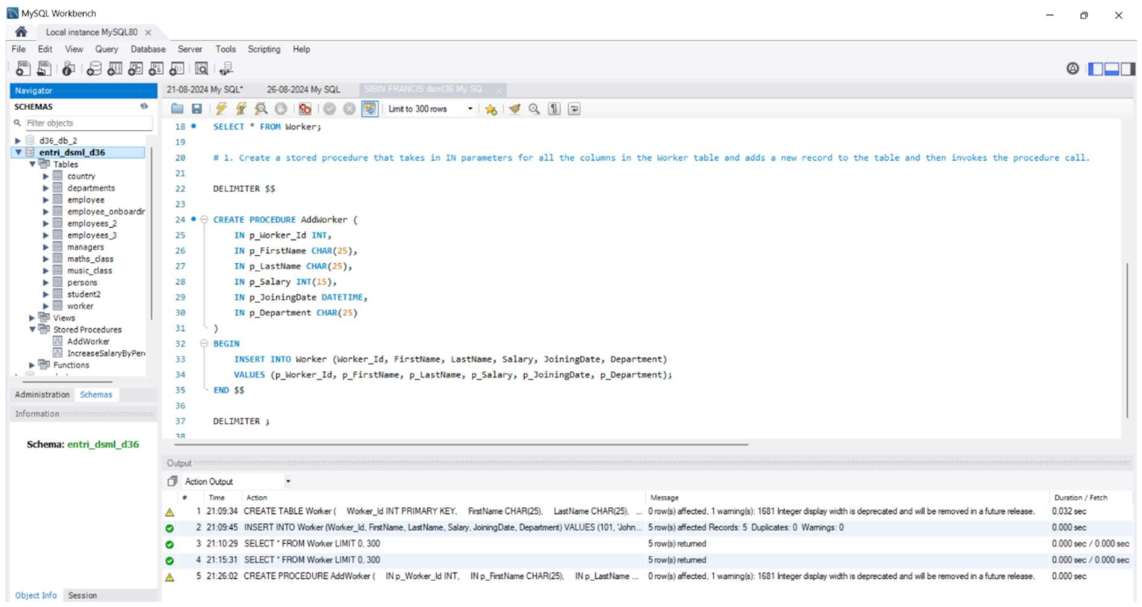
➢ CREATED TABLE worker



```sql
CREATE TABLE Worker (
    Worker_Id INT PRIMARY KEY,
    FirstName CHAR(25),
    LastName CHAR(25),
    Salary INT(15),
    JoiningDate DATETIME,
    Department CHAR(25)
);

INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)
VALUES
(101, 'John', 'Doe', 50000, '2022-01-15', 'HR'),
(102, 'Jane', 'Smith', 60000, '2021-03-10', 'Finance'),
(103, 'Michael', 'Johnson', 55000, '2020-11-20', 'IT'),
(104, 'Emily', 'Davis', 48000, '2019-07-05', 'HR'),
(105, 'Robert', 'Brown', 70000, '2023-06-12', 'Marketing');

SELECT * FROM Worker;
```

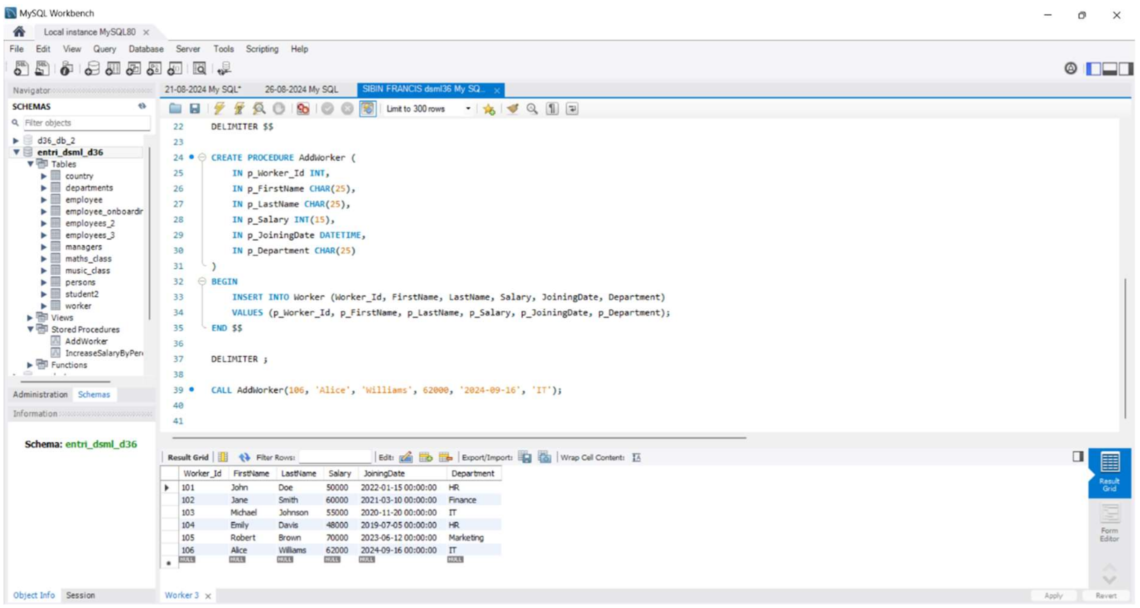
Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
101	John	Doe	50000	2022-01-15 00:00:00	HR
102	Jane	Smith	60000	2021-03-10 00:00:00	Finance
103	Michael	Johnson	55000	2020-11-20 00:00:00	IT
104	Emily	Davis	48000	2019-07-05 00:00:00	HR
105	Robert	Brown	70000	2023-06-12 00:00:00	Marketing
NULL	NULL	NULL	NULL	NULL	NULL

➢ CREATED A STORED PROCEDURE AddWorker



```sql
SELECT * FROM Worker;

# 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 AddWorker (
    IN p_Worker_Id INT,
    IN p_FirstName CHAR(25),
    IN p_LastName CHAR(25),
    IN p_Salary INT(15),
    IN p_JoiningDate DATETIME,
    IN p_Department CHAR(25)
)
BEGIN
    INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)
    VALUES (p_Worker_Id, p_FirstName, p_LastName, p_Salary, p_JoiningDate, p_Department);
END $$

DELIMITER ;
```

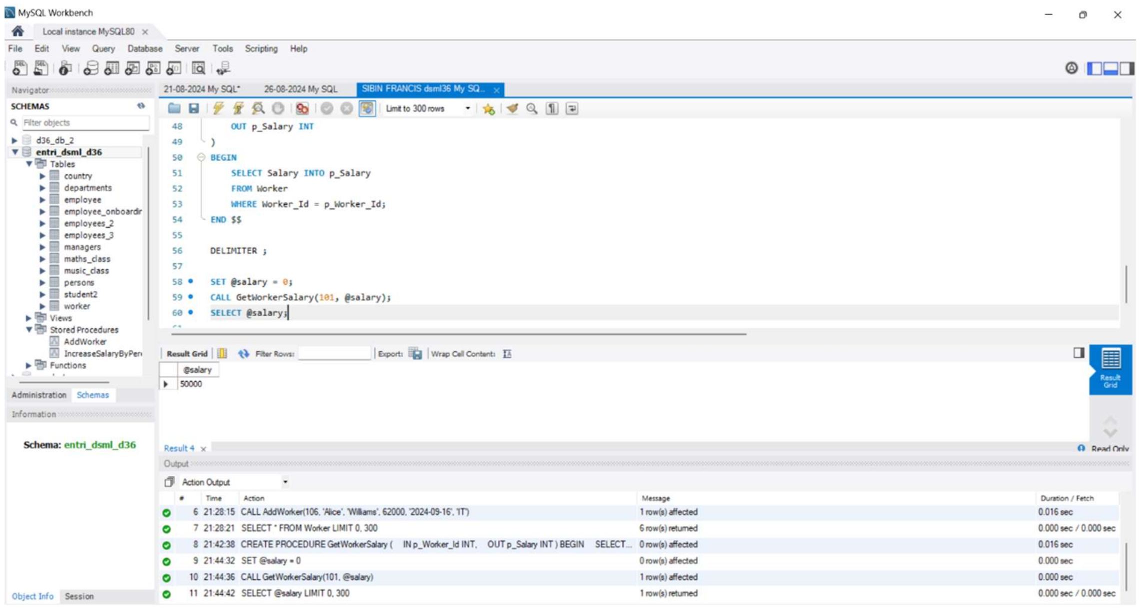
#	Time	Action	Message	Duration / Fetch
1	21:09:34	CREATE TABLE Worker (Worker_Id INT PRIMARY KEY, FirstName CHAR(25), LastName CHAR(25), ...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.032 sec
2	21:09:45	INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department) VALUES (101, 'John...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
3	21:10:29	SELECT * FROM Worker LIMIT 0, 300	5 row(s) returned	0.000 sec / 0.000 sec
4	21:15:31	SELECT * FROM Worker LIMIT 0, 300	5 row(s) returned	0.000 sec / 0.000 sec
5	21:26:02	CREATE PROCEDURE AddWorker (IN p_Worker_Id INT, IN p_FirstName CHAR(25), IN p_LastName ...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.000 sec



```
DELIMITER $$

CREATE PROCEDURE AddWorker (
    IN p_Worker_Id INT,
    IN p_FirstName CHAR(25),
    IN p_LastName CHAR(25),
    IN p_Salary INT(15),
    IN p_JoiningDate DATETIME,
    IN p_Department CHAR(25)
)
BEGIN
    INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)
    VALUES (p_Worker_Id, p_FirstName, p_LastName, p_Salary, p_JoiningDate, p_Department);
END $$

DELIMITER ;

CALL AddWorker(106, 'Alice', 'Williams', 62000, '2024-09-16', 'IT');
```

Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
101	John	Doe	50000	2022-01-15 00:00:00	HR
102	Jane	Smith	60000	2021-03-10 00:00:00	Finance
103	Michael	Johnson	55000	2020-11-20 00:00:00	IT
104	Emily	Davis	48000	2019-07-05 00:00:00	HR
105	Robert	Brown	70000	2023-06-12 00:00:00	Marketing
106	Alice	Williams	62000	2024-09-16 00:00:00	IT
NULL	NULL	NULL	NULL	NULL	NULL

➢ **CREATED A STORED PROCEDURE GetWorkerSalary**

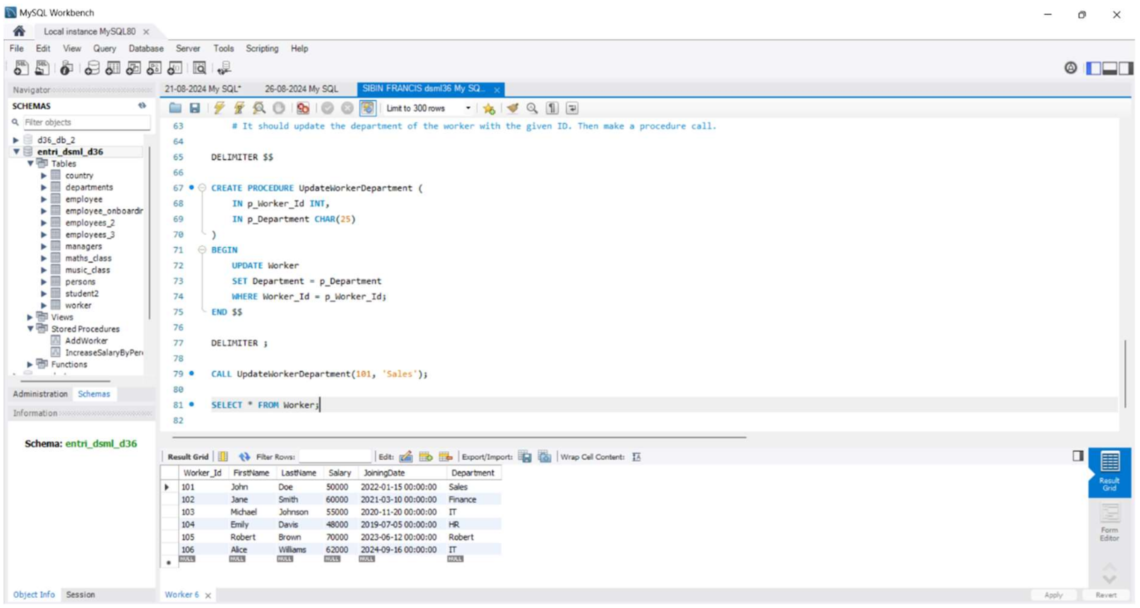


```
    OUT p_Salary INT
)
BEGIN
    SELECT Salary INTO p_Salary
    FROM Worker
    WHERE Worker_Id = p_Worker_Id;
END $$

DELIMITER ;

SET @salary = 0;
CALL GetWorkerSalary(101, @salary);
SELECT @salary;
```

@salary
50000

```
 #   Time      Action                                                                              Message            Duration / Fetch
 6   21:28:15  CALL AddWorker(106, 'Alice', 'Williams', 62000, '2024-09-16', 'IT')                 1 row(s) affected  0.016 sec
 7   21:28:21  SELECT * FROM Worker LIMIT 0, 300                                                   6 row(s) returned  0.000 sec / 0.000 sec
 8   21:42:38  CREATE PROCEDURE GetWorkerSalary (    IN p_Worker_Id INT,    OUT p_Salary INT ) BEGIN    SELECT...  0 row(s) affected  0.016 sec
 9   21:44:32  SET @salary = 0                                                                     0 row(s) affected  0.000 sec
10   21:44:36  CALL GetWorkerSalary(101, @salary)                                                  1 row(s) affected  0.000 sec
11   21:44:42  SELECT @salary LIMIT 0, 300                                                         1 row(s) returned  0.000 sec / 0.000 sec
```

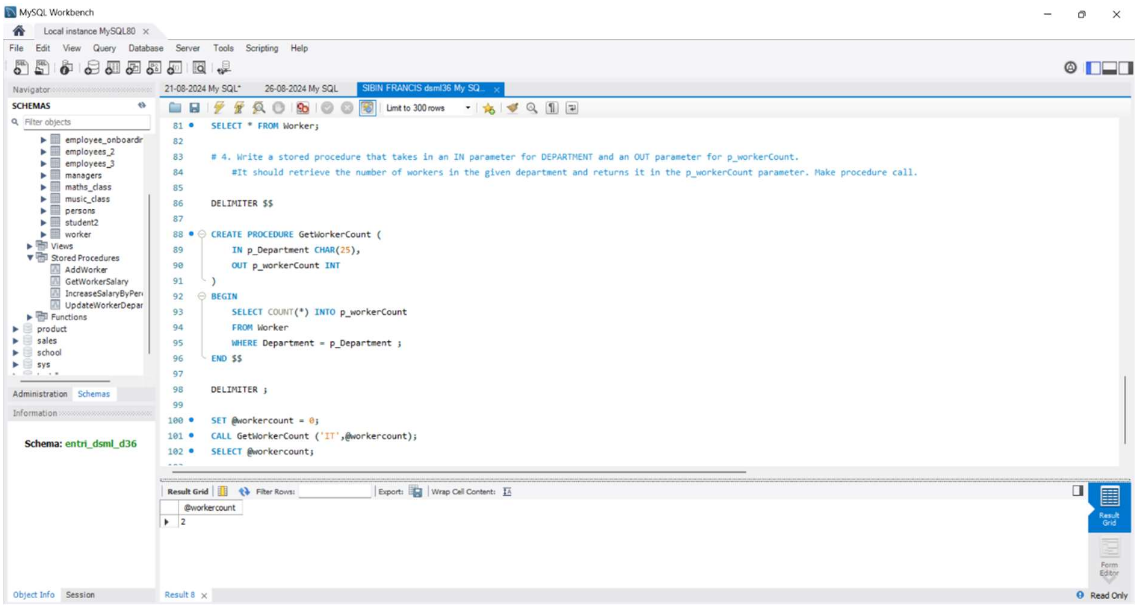
- **CREATED A STORED PROCEDURE UpdateWorkerDepartment**



```sql
    # It should update the department of the worker with the given ID. Then make a procedure call.

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 ;

CALL UpdateWorkerDepartment(101, 'Sales');

SELECT * FROM Worker;
```

Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
101	John	Doe	50000	2022-01-15 00:00:00	Sales
102	Jane	Smith	60000	2021-03-10 00:00:00	Finance
103	Michael	Johnson	55000	2020-11-20 00:00:00	IT
104	Emily	Davis	48000	2019-07-05 00:00:00	HR
105	Robert	Brown	70000	2023-06-12 00:00:00	Robert
106	Alice	Williams	62000	2024-09-16 00:00:00	IT
NULL	NULL	NULL	NULL	NULL	NULL

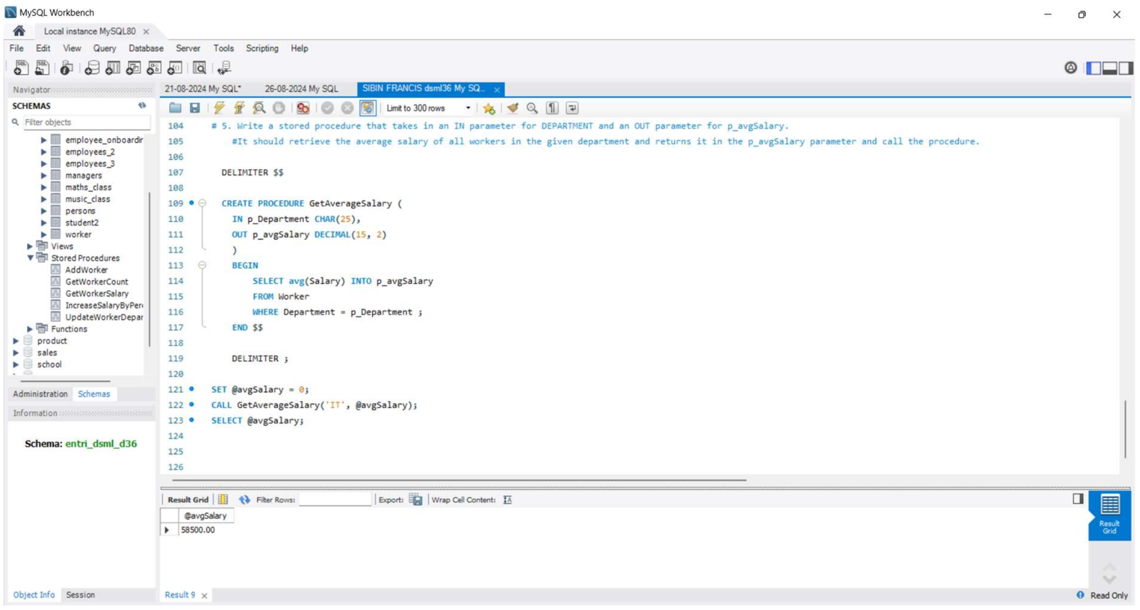
- **CREATED A STORED PROCEDURE GetWorkerCount**



```sql
SELECT * FROM Worker;

# 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 GetWorkerCount (
    IN p_Department CHAR(25),
    OUT p_workerCount INT
)
BEGIN
    SELECT COUNT(*) INTO p_workerCount
    FROM Worker
    WHERE Department = p_Department ;
END $$

DELIMITER ;

SET @workercount = 0;
CALL GetWorkerCount ('IT',@workercount);
SELECT @workercount;
```

@workercount
2

➢ CREATED A STORED PROCEDURE GetAverageSalary



```sql
# 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 GetAverageSalary (
   IN p_Department CHAR(25),
   OUT p_avgSalary DECIMAL(15, 2)
   )
   BEGIN
       SELECT avg(Salary) INTO p_avgSalary
       FROM Worker
       WHERE Department = p_Department ;
   END $$

   DELIMITER ;

SET @avgSalary = 0;
CALL GetAverageSalary('IT', @avgSalary);
SELECT @avgSalary;
```

```
@avgSalary
58500.00
```