

STORED PROCEDURES

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.

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left shows the 'Schemas' tree with 'worker' selected. The 'SQL File 9*' editor shows the following SQL code:

```
1 delimiter $$
2 create procedure entry(in id int,in fname char(25),in lname char(25),in sal int,in jdate datetime,in dept char(25))
3 begin
4 insert into worker values(id,fname,lname,sal,jdate,dept);
5 end $$
6 delimiter ;
7 call entry(1, 'John', 'Doe', 50000, '2020-01-15 09:00:00', 'HR');
8 call entry(2, 'Jane', 'Smith', 60000, '2019-02-23 10:30:00', 'Finance');
9 call entry(3, 'Mike', 'Johnson', 45000, '2021-03-12 11:00:00', 'IT');
10 call entry(4, 'Emily', 'Davis', 70000, '2018-07-11 08:45:00', 'Sales');
11 select * from worker;
12
```

The 'Result Grid' shows the output of the 'select * from worker' query:

Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
1	John	Doe	50000	2020-01-15 09:00:00	HR
2	Jane	Smith	60000	2019-02-23 10:30:00	Finance
3	Mike	Johnson	45000	2021-03-12 11:00:00	IT
4	Emily	Davis	70000	2018-07-11 08:45:00	Sales

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.

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left shows the 'Schemas' tree with 'worker' selected. The 'SQL File 9*' editor shows the following SQL code:

```
13 delimiter $$
14 create procedure retrievalsal(in id int,out p_salary int)
15 begin
16 select Salary into p_salary from worker where Worker_Id=id;
17 end $$
18 delimiter ;
19 call retrievalsal(1,@s);
20 select @s as Salary_of_worker;
```

The 'Result Grid' shows the output of the 'select @s as Salary_of_worker' query:

Salary_of_worker
50000

The 'Column: Worker_Id' pane shows the definition: 'Worker_Id int'.

The 'Result 5' pane shows the output of the 'call retrievalsal(1,@s)' query:

#	Time	Action	Message	Duration / Fetch
12	18:58:07	select @s as Salary_of_worker LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
13	18:58:12	select * from worker LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	18:58:18	select @s as Salary_of_worker LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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.

The screenshot shows MySQL Workbench with a SQL script in the editor. The script creates a stored procedure named `updatedept1` that takes two IN parameters: `in id int` and `in dept char(25)`. The procedure updates the `Department` of the worker with the given ID. Below the script, the `Result Grid` shows the execution results. The `worker` table is displayed with columns: `Worker_Id`, `FirstName`, `LastName`, `Salary`, `JoiningDate`, and `Department`. The output shows that the department of worker 3 was updated from 'Marketing' to 'Marketing'.

```
23 delimiter $$
24 create procedure updatedept1(in id int,in dept char(25))
25 begin
26 update worker set Department=dept where Worker_Id=id;
27 end $$
28 delimiter ;
29 call updatedept1(3,'Marketing');
30 select * from worker where Worker_Id=3;
31
```

Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
3	Mike	Johnson	45000	2021-03-12 11:00:00	Marketing

Column: **Worker_Id**
Definition: Worker_Id int

#	Time	Action	Message	Duration / Fetch
21	19:01:58	set sql_safe_updates=0	0 row(s) affected	0.000 sec
22	19:02:01	call updatedept1(3,'Marketing')	1 row(s) affected	0.000 sec
23	19:02:24	select * from worker where Worker_Id=3 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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.

The screenshot shows MySQL Workbench with a SQL script in the editor. The script creates a stored procedure named `countdept2` that takes an IN parameter `in dept char(25)` and an OUT parameter `out p_worker_count int`. The procedure counts the number of workers in the given department and returns the count. Below the script, the `Result Grid` shows the execution results. The `Workers_Count` table is displayed with columns: `Workers_Count`. The output shows that the count of workers in the 'Finance' department is 2.

```
33
34 delimiter $$
35 create procedure countdept2(in dept char(25),out p_worker_count int)
36 begin
37 select count(Department) as Workers_Count from worker where Department=dept;
38 end $$
39 delimiter ;
40 call countdept2('Finance',@c);
41
```

Workers_Count
2

Column: **Department**
Collation: utf8mb4_0900_ai_ci
Definition: Department char(25)

#	Time	Action	Message	Duration / Fetch
29	19:06:51	call countdept1('Finance',@c)	1 row(s) returned	0.000 sec / 0.000 sec
30	19:07:26	create procedure countdept2(in dept char(25),out p_worker_count int) begin select co...	0 row(s) affected	0.015 sec
31	19:07:27	call countdept2('Finance',@c)	1 row(s) returned	0.000 sec / 0.000 sec

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.

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following script:

```
42 end $$
43 delimiter ;
44 call countdept2('Finance',@c);
45
46 select * from worker;
47
```

The Result Grid displays the output of the script, showing a table with 10 rows of worker data:

Worker_Id	Firstname	Lastname	Salary	JoiningDate	Department
1	John	Doe	50000	2020-01-15 09:00:00	HR
2	Jane	Smith	60000	2019-02-23 10:30:00	Finance
3	Mike	Johnson	45000	2021-03-12 11:00:00	Marketing
4	Emily	Davis	70000	2018-07-11 08:45:00	Sales
5	David	Lee	53000	2019-10-05 09:30:00	Finance
6	Anna	Wilson	62000	2020-12-20 10:45:00	IT
7	Chris	Brown	55000	2017-09-14 09:15:00	Marketing
8	Sophia	Martinez	49000	2021-01-25 08:00:00	HR
9	James	Taylor	58000	2016-04-12 07:30:00	Sales
10	Olivia	Harris	61000	2019-08-16 11:15:00	Marketing

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following script:

```
48
49 delimiter $$
50 create procedure AvgSalary1(in dept char(25),out p_avgsalary int)
51 begin
52 select Department,avg(Salary) as AverageSalary from worker group by Department having Department=dept;
53 end $$
54 delimiter ;
55 call AvgSalary1('IT',@av);
56
```

The Result Grid displays the output of the script, showing a table with 2 rows of average salary data:

Department	AverageSalary
IT	62000.0000

The Output pane shows the execution of the stored procedure:

```
42 19:13:10 call AvgSalary1(IT,@av) 1 row(s) returned 0.000 sec / 0.000 sec
43 19:13:20 call AvgSalary1(IT,@av) 1 row(s) returned 0.000 sec / 0.000 sec
```