# **Hands-on Lab: Stored Procedures**



Estimated time needed: 20 minutes

Stored Procedures in SQL are a type of database object that allow you to encapsulate a series of SQL statements into a single routine. They are stored in the database data dictionary and can be invoked from an application program or from the database command interface. Stored procedures can accept input parameters and return multiple values of output parameters. They can also include control-of-flow constructs such as loops and conditional statements. Stored procedures offer several benefits including improved performance, higher productivity, ease of use, and increased scalability. They also provide a mechanism for enforcing business rules and data integrity in the database system.

### **Objectives**

After completing this lab, you will be able to:

- Create stored procedures
- · Execute stored procedures

### Software Used in this Lab

In this lab, you will use MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.



To complete this lab you will utilize MySQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

#### **Database Used in this Lab**

Mysql\_learners database has been used in this lab.

### **Data Used in this Lab**

The data used in this lab is internal data. You will be working on the PETSALE table.

ID 🛋	ANIMAL	SALEPRICE
1	Cat	450.09
2	Dog	666.66
3	Parrot	50.00
4	Hamster	60.60
5	Goldfish	48.48

This lab requires you to have the PETSALE table populated with sample data on mysql phpadmin interface. You might have created and populated a PETSALE table in a previous lab.

about:blank 1/10

For this lab, you need to create a database PETS in the phpMyAdmin interface. Download the PETSALE-CREATE-v2.sql script below, upload it to console under the PETS database. Upon execution, the script will create a new PETSALE table dropping any previous PETSALE table if exists, and will populate it with the required sample data.

• PETSALE-CREATE-v2.sql

1. 1

### **Stored Procedure: Exercise 1**

In this exercise, you will create and execute a stored procedure to read data from a table on mysql phpadmin using SQL.

- 1. You will create a stored procedure routine named **RETRIEVE ALL**.
  - This RETRIEVE\_ALL routine will contain an SQL query to retrieve all the records from the PETSALE table, so you don't need to write the same query
    over and over again. You just call the stored procedure routine to execute the query everytime.
  - To create the stored procedure routine, copy the code below and paste it to the textarea of the SQL page. Click Go.

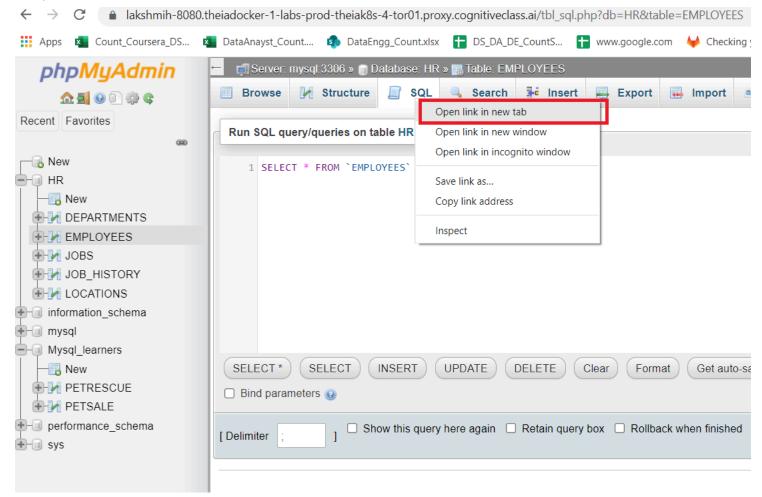
```
2. 2
3. 3
 8.8
 1. DELIMITER //
 3. CREATE PROCEDURE RETRIEVE_ALL()
 5. BEGIN
      SELECT * FROM PETSALE;
 7. END //
 8. DELIMITER;
Copied!
      Run SQL query/queries on database Mysql_learners: 🔞
          1 DELIMITER //
          2
            CREATE PROCEDURE RETRIEVE_ALL()
          4
            BEGIN
          6
                 SELECT * FROM PETSALE;
          8
        10 END //
        11
        12 DELIMITER;
       Clear
                Format
                            Get auto-saved query
     ☐ Bind parameters 

    Show this query here again
    Retain query box
    Rollback when finished
    Enable foreign key checks

    [ Delimiter
    Hide query box
      MySQL returned an empty result set (i.e. zero rows). (Query took 0.0064 seconds.)
     CREATE PROCEDURE RETRIEVE_ALL() BEGIN SELECT * FROM PETSALE; END
```

2. To call the RETRIEVE\_ALL routine, open another SQL tab by clicking Open in new Tab

about:blank 2/10



Delete the default line which appears so that you will get a blank window.

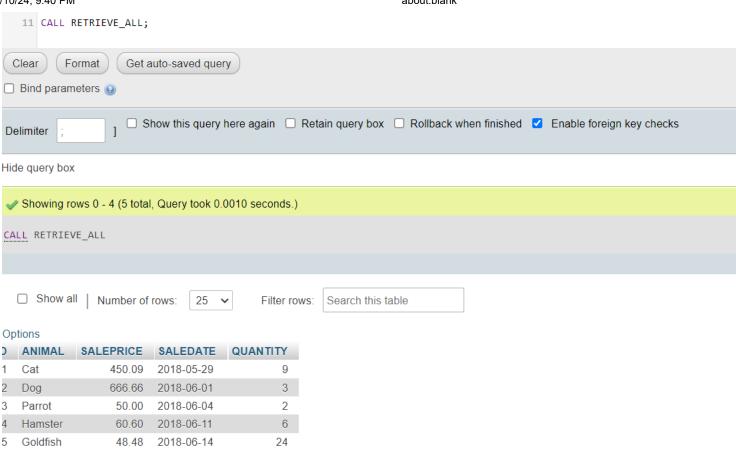
Copy the code below and paste it to the textarea of the SQL page. Click Go.

1. 1

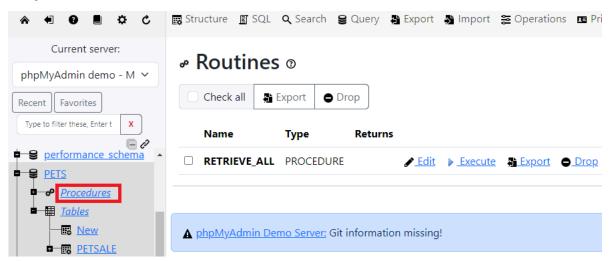
CALL RETRIEVE\_ALL;

Copied!

about:blank 3/10



3. You can view the created stored procedure routine RETRIEVE ALL. On the left panel, expand the PETS database option and click on Procedures to view the procedure.



4. If you wish to drop the stored procedure routine RETRIEVE\_ALL, copy the code below and paste it to the textarea of the SQL page. Click Go.

1. 1

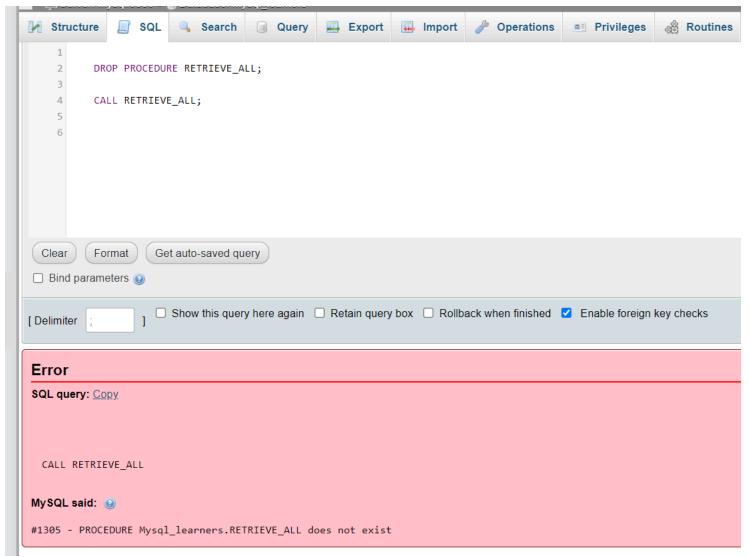
2. 2 3. 3

DROP PROCEDURE RETRIEVE\_ALL;

2.3. CALL RETRIEVE\_ALL;

Copied!

about:blank 4/10



# **Stored Procedure: Exercise 2**

In this exercise, you will create and execute a stored procedure to write/modify data in a table on MySQL using SQL.

You will create a stored procedure routine named UPDATE SALEPRICE with parameters Animal ID and Animal Health.

- This UPDATE\_SALEPRICE routine will contain SQL queries to update the sale price of the animals in the PETSALE table depending on their health conditions, BAD or WORSE.
- This procedure routine will take animal ID and health condition as parameters which will be used to update the sale price of animal in the PETSALE table by an amount depending on their health condition. Suppose that:
  - $\circ~$  For animal with ID XX having BAD health condition, the sale price will be reduced further by 25%.
  - $\circ$  For animal with ID YY having WORSE health condition, the sale price will be reduced further by 50%.
  - For animal with ID ZZ having other health condition, the sale price won't change.

• To create the stored procedure routine, copy the code below and paste it to the textarea of the SQL page. Click Go.

1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 12. 12 13. 13 14. 14 15. 15 16. 16 17. 17

18. 18

about:blank 5/10

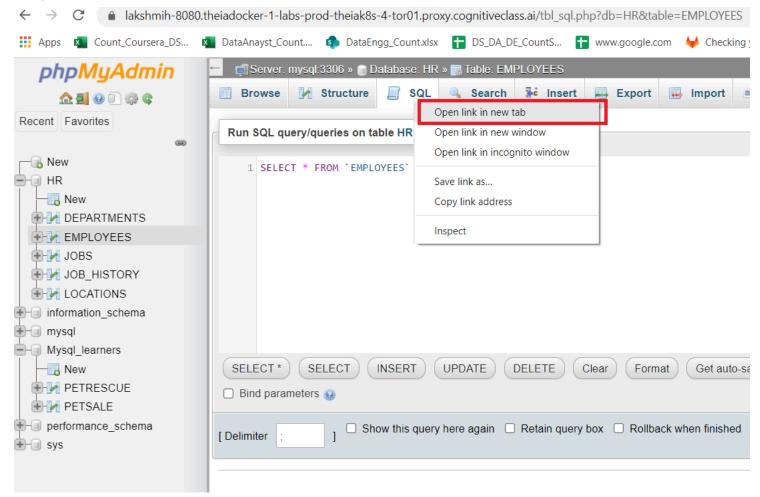
```
1. DELIMITER @
 2. CREATE PROCEDURE UPDATE_SALEPRICE (IN Animal_ID INTEGER, IN Animal_Health VARCHAR(5))
 3. BEGIN
       IF Animal_Health = 'BAD' THEN
    UPDATE PETSALE
 4.
 5.
          SET SALEPRICE = SALEPRICE - (SALEPRICE * 0.25)
 6.
          WHERE ID = Animal_ID;
 7.
 8.
       ELSEIF Animal_Health = 'WORSE' THEN
 9.
          UPDATE PETSALE
          SET SALEPRICE = SALEPRICE - (SALEPRICE * 0.5)
10.
          WHERE ID = Animal_ID;
11.
12.
       ELSE
          UPDATE PETSALE
13.
          SET SALEPRICE = SALEPRICE
15.
          WHERE ID = Animal_ID;
       END IF;
16.
17. END @
18.
19. DELIMITER;
Copied!
 Structure
                SQL
                               Search
                                            Query
                                                     Export
                                                                  - Import
                                                                                Operations
                                                                                                  Privileges
                                                                                                                  Routines 8
  Run SQL query/queries on database Mysql_learners:
    15
    16
            ELSE
    17
                UPDATE PETSALE
                SET SALEPRICE = SALEPRICE
    18
                WHERE ID = Animal_ID;
    19
    20
    21
            END IF;
    22
    23 END @
    24
    25 DELIMITER;
    26
  Clear
            Format
                       Get auto-saved query

    Bind parameters (a)

                       🗌 Show this query here again 🔲 Retain query box 🔲 Rollback when finished 🗹 Enable foreign key checks
[ Delimiter
Hide query box
 CREATE PROCEDURE UPDATE_SALEPRICE ( IN Animal_ID INTEGER, IN Animal_Health VARCHAR(5) ) BEGIN IF Animal_Health =
 (SALEPRICE * 0.25) WHERE ID = Animal_ID; ELSEIF Animal_Health = 'WORSE' THEN UPDATE PETSALE SET SALEPRICE = SALEPRICE
 PETSALE SET SALEPRICE = SALEPRICE WHERE ID = Animal_ID; END IF; END
```

1. Let's call the UPDATE\_SALEPRICE routine. We want to update the sale price of animal with ID 1 having BAD health condition in the PETSALE table. open another SQL tab by clicking Open in new Tab

about:blank 6/10



Delete the default line which appears so that you will get a blank window.

Copy the code below and paste it to the textarea of the SQL page. Click Go.

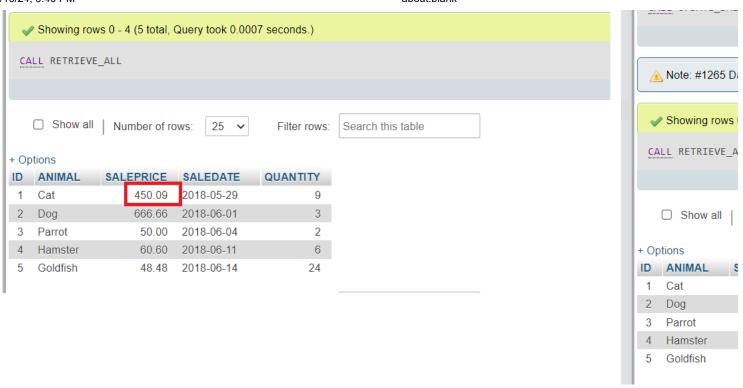
Note if you have dropped RETREIVE\_ALL procedure rerun the creation script of that procedure before executing these lines.

```
1. 1
2. 2
3. 3
4. 4
5. 5

1. CALL RETRIEVE_ALL;
2.
3. CALL UPDATE_SALEPRICE(1, 'BAD');
4.
5. CALL RETRIEVE_ALL;

Copied!
```

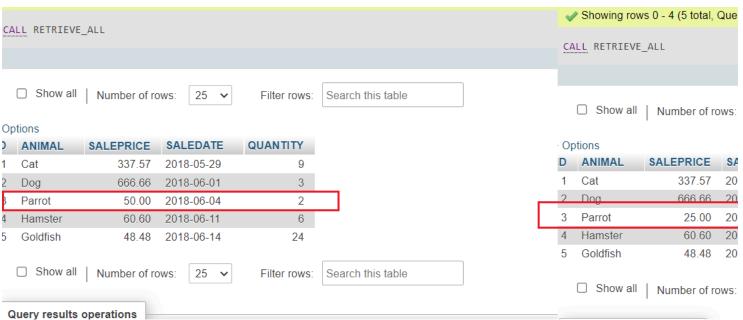
about:blank 7/10



2. Let's call the UPDATE\_SALEPRICE routine once again. We want to update the sale price of animal with ID 3 having WORSE health condition in the PETSALE table. copy the code below and paste it to the textarea of the SQL page. Click Go. You will have all the records retrieved from the PETSALE table.

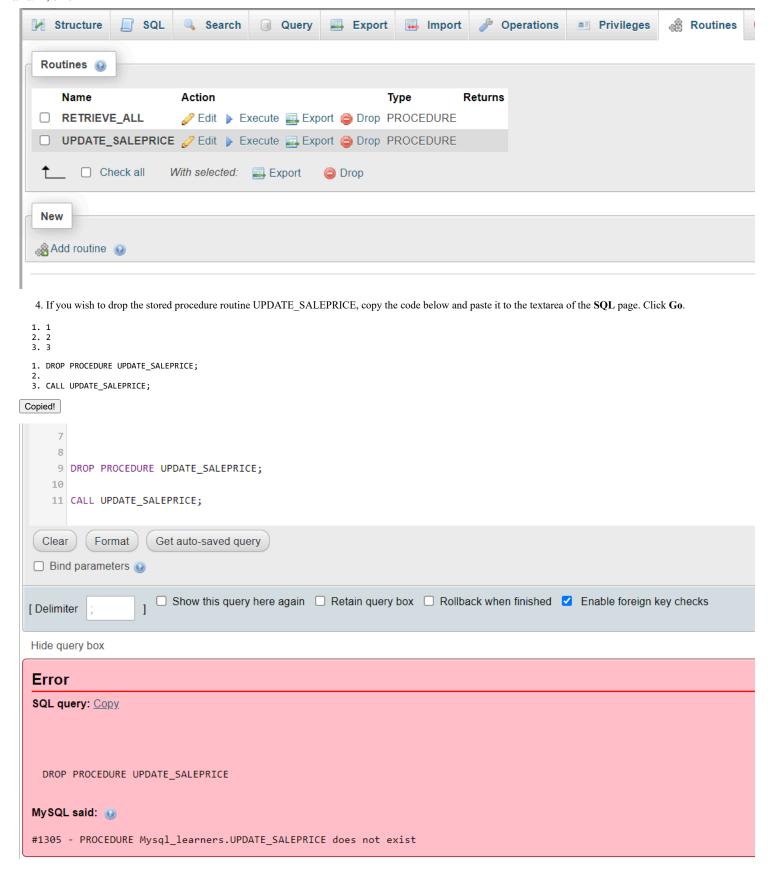
```
1. 1
2. 2
3. 3
4. 4
5. 5

1. CALL RETRIEVE_ALL;
2.
3. CALL UPDATE_SALEPRICE(3, 'WORSE');
4.
5. CALL RETRIEVE_ALL;
Copied!
```



3. You can view the created stored procedure routine UPDATE\_SALEPRICE. Click on the **Routines** and view the procedure.

about:blank 8/10



# **Conclusion**

Congratulations! You have completed this lab on creating stored procedures in MySQL.

You are now able to:

- Write a stored procedure as per requirement
- Call or Exectue a stored procedure

about:blank 9/10

• Drop a stored procedure once its utility is over

## Author(s)

<u>Lakshmi Holla</u>

Malika Singla

Abhishek Gagneja

# Changelog

Date	Version	Changed by	Change Description
2023-10-31	0.4	Mercedes Schneider	QA Edits
2023-10-16	0.3	Abhishek Gagneja	Updated the instructions
2021-08-09	0.2	Sathya Priya	Updated HTML tags and SQL link
2021-11-01	0.1	Lakshmi Holla, Malika Singla	Initial Version

© IBM Corporation 2023. All rights reserved.

about:blank