

# Hands-on Lab: Stored Procedures in MySQL using phpMyAdmin

**Estimated time needed:** 20 minutes

In this lab, you will learn how to create tables and load data in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

## Software Used in this Lab

In this lab, you will use [MySQL](#). 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. But for this lab, it is recommended you download the **PETSALE-CREATE-v2.sql** script below, upload it to phpadmin console and run it. 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](#)

## Objectives

After completing this lab, you will be able to:

- Create stored procedures
- Execute stored procedures

## 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. Make sure you have created and populated the **PETSALE** table following the steps in the “**Data Used in this Lab**” section of this lab.

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

- 2.
- 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**.

```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12

1. DELIMITER //
2.
3. CREATE PROCEDURE RETRIEVE_ALL()
4.
5. BEGIN
6.
7.     SELECT * FROM PETSale;
8.
9.
10. END //
11.
12. DELIMITER ;

```

Copied!

Run SQL query/queries on database Mysql\_learners: ?

```
1 DELIMITER //
2
3 CREATE PROCEDURE RETRIEVE_ALL()
4
5 BEGIN
6
7     SELECT * FROM PETALE;
8
9
10 END //
11
12 DELIMITER ;
```

Clear

Format

Get auto-saved query

☐ Bind parameters ?

[ Delimiter ; ] ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

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 PETALE; END
```

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

← → ↻ 🔒 lakshmih-8080.theiadocker-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/tbl\_sql.php?db=HR&table=EMPLOYEES

Apps Count\_Coursera\_DS... DataAnayst\_Count... DataEngg\_Count.xlsx DS\_DA\_DE\_CountS... www.google.com Checking y

**phpMyAdmin**

Recent Favorites

- New
- HR
  - New
  - DEPARTMENTS
  - EMPLOYEES
  - JOBS
  - JOB\_HISTORY
  - LOCATIONS
- information\_schema
- mysql
- Mysql\_learners
  - New
  - PETRESCUE
  - PETSALE
- performance\_schema
- sys

Server: mysql:3306 » Database: HR » Table: EMPLOYEES

Browse Structure SQL Search Insert Export Import

Run SQL query/queries on table HR

1 SELECT \* FROM `EMPLOYEES`

Open link in new tab  
Open link in new window  
Open link in incognito window  
Save link as...  
Copy link address  
Inspect

SELECT \* SELECT INSERT UPDATE DELETE Clear Format Get auto-se

☐ Bind parameters ⓘ

[ Delimiter ; ] ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished

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  
1. CALL RETRIEVE_ALL;
```

Copied!

11 CALL RETRIEVE\_ALL;

Clear

Format

Get auto-saved query

☐ Bind parameters ?

Delimiter ; ]

☐ Show this query here again

☐ Retain query box

☐ Rollback when finished

☒ Enable foreign key checks

Hide query box

✓ Showing rows 0 - 4 (5 total, Query took 0.0010 seconds.)

CALL RETRIEVE\_ALL

☐ Show all

Number of rows:

25

Filter rows:

Search this table

Options

	ANIMAL	SALEPRICE	SALEDATE	QUANTITY
1	Cat	450.09	2018-05-29	9
2	Dog	666.66	2018-06-01	3
3	Parrot	50.00	2018-06-04	2
4	Hamster	60.60	2018-06-11	6
5	Goldfish	48.48	2018-06-14	24

4. You can view the created stored procedure routine RETRIEVE\_ALL. On the left panel, expand the mysql option. Click on **Procedures** then click on the **RETRIEVE\_ALL** and view the procedure.

The screenshot shows the phpMyAdmin interface. On the left sidebar, the 'mysql' database is expanded, and 'Procedures' is selected. The 'RETRIEVE\_ALL' procedure is highlighted. The main panel shows the procedure definition: `DROP PROCEDURE `RETRIEVE_ALL`; CREATE DEFINER=`root`@`%` PROCEDURE `RETRIEVE_ALL`() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER FROM PETSALE; END`. Below the definition, there is a text area for running SQL queries on the table 'mysql.PETSALE'. The query `SELECT * FROM `PETSALE` WHERE 1` is entered. The results of the query are displayed in a table with columns: ID, ANIMAL, SALEPRICE, SALEDATE, and QUANTITY. The results show 5 rows of data.

Server: mysql:3306 » Database: mysql » Table: PETSALE

Browse Structure SQL Search Insert Export Import Privileges Operations

✓ Routine 'RETRIEVE\_ALL' has been modified.

DROP PROCEDURE `RETRIEVE\_ALL`; CREATE DEFINER=`root`@`%` PROCEDURE `RETRIEVE\_ALL`() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER FROM PETSALE; END

Run SQL query/queries on table mysql.PETSALE: ?

1 SELECT \* FROM `PETSALE` WHERE 1

Columns

ID	ANIMAL	SALEPRICE	SALEDATE	QUANTITY
1	Cat	450.09	2018-05-29	9
2	Dog	666.66	2018-06-01	3
3	Parrot	50.00	2018-06-04	2
4	Hamster	60.60	2018-06-11	6
5	Goldfish	48.48	2018-06-14	24

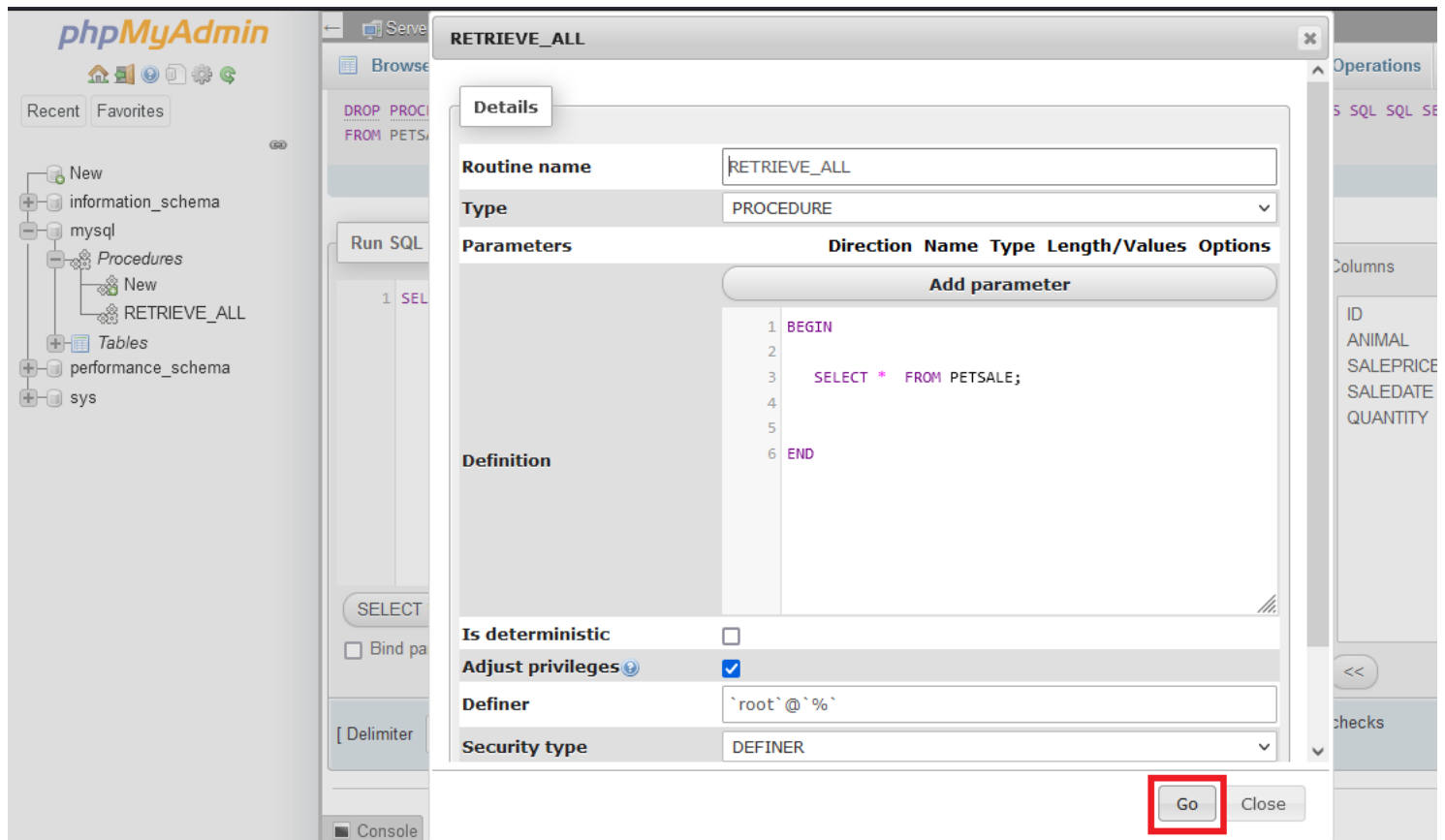
SELECT \* SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters ?

Delimiter ; ] ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Console

After clicking on the Procedure **Retrieve\_All**, you can view the procedure definition and execute it by clicking on **GO**.



5. 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
```

```
1. DROP PROCEDURE RETRIEVE_ALL;
2.
3. CALL RETRIEVE_ALL;
```

Copied!

Structure

SQL

Search

Query

Export

Import

Operations

Privileges

Routines

1

2

3

4

5

6

DROP PROCEDURE RETRIEVE\_ALL;

CALL RETRIEVE\_ALL;

Clear

Format

Get auto-saved query

☐ Bind parameters

[ Delimiter ; ]

☐ Show this query here again

☐ Retain query box

☐ Rollback when finished

☒ Enable foreign key checks

Error

SQL query: [Copy](#)

CALL RETRIEVE\_ALL

MySQL said:

#1305 - PROCEDURE Mysql\_learners.RETRIEVE\_ALL does not exist

## Exercise 2

In this exercise, you will create and execute a stored procedure to write/modify data in a table on Db2 using SQL.

1. Make sure you have created and populated the **PETSALE** table following the steps in the “Data Used in this Lab” section of this lab.

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

- 2.
- 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 -
    - For animal with ID XX having BAD health condition, the sale price will be reduced further by 25%.
    - 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
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26

1. DELIMITER @
2. CREATE PROCEDURE UPDATE_SALEPRICE (
3.     IN Animal_ID INTEGER, IN Animal_Health VARCHAR(5) )
4. BEGIN
5.
6.     IF Animal_Health = 'BAD' THEN
7.         UPDATE PETSale
8.         SET SALEPRICE = SALEPRICE - (SALEPRICE * 0.25)
9.         WHERE ID = Animal_ID;
10.
11.     ELSEIF Animal_Health = 'WORSE' THEN
12.         UPDATE PETSale
13.         SET SALEPRICE = SALEPRICE - (SALEPRICE * 0.5)
14.         WHERE ID = Animal_ID;
15.
16.     ELSE
17.         UPDATE PETSale
18.         SET SALEPRICE = SALEPRICE
19.         WHERE ID = Animal_ID;
20.
21.     END IF;
22.
23. END @
24.
25. DELIMITER ;
26.
```

Copied!



Server: MySQL 5.6.27 Database: mysql\_learners

Structure SQL Search Query Export Import Operations Privileges Routines

Run SQL query/queries on database Mysql\_learners:

```
15
16 ELSE
17     UPDATE PETALE
18     SET SALEPRICE = SALEPRICE
19     WHERE ID = Animal_ID;
20
21 END IF;
22
23 END @
24
25 DELIMITER ;
26
```

Clear Format Get auto-saved query

☐ Bind parameters

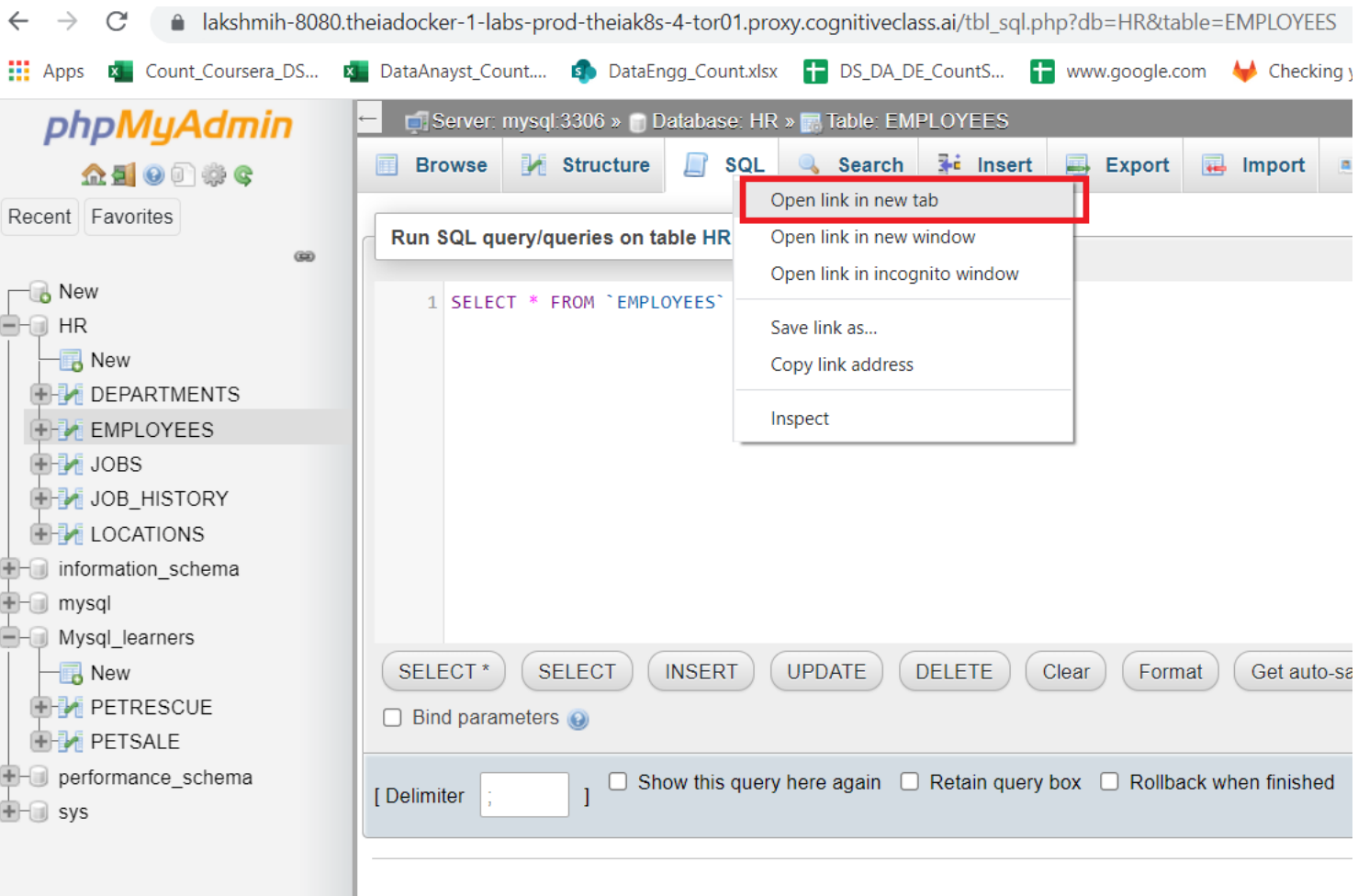
[ Delimiter ; ] ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Hide query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0214 seconds.)

```
CREATE PROCEDURE UPDATE_SALEPRICE ( IN Animal_ID INTEGER, IN Animal_Health VARCHAR(5) ) BEGIN IF Animal_Health = 'BAD' THEN
(SALEPRICE * 0.25) WHERE ID = Animal_ID; ELSEIF Animal_Health = 'WORSE' THEN UPDATE PETALE SET SALEPRICE = SALEPRICE - (
PETALE SET SALEPRICE = SALEPRICE WHERE ID = Animal_ID; END IF; END
```

- 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 PETALE table. open another SQL tab by clicking **Open in new Tab**



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!



Structure

SQL

Search

Query

Export

Import

Operations

Privileges

Routines

Routines

Name	Action	Type	Returns
<input type="checkbox"/> RETRIEVE_ALL	Edit  Execute  Export  Drop	PROCEDURE	
<input type="checkbox"/> UPDATE_SALEPRICE	Edit  Execute  Export  Drop	PROCEDURE	

☐ Check all
 With selected: Export Drop

New

Add routine

6. If you wish to drop the stored procedure routine UPDATE\_SALEPRICE, copy the code below and paste it to the textarea of the **SQL** page. Click **Go**.

```

1. 1
2. 2
3. 3

1. DROP PROCEDURE UPDATE_SALEPRICE;
2.
3. CALL UPDATE_SALEPRICE;
```

Copied!

7

8

9 DROP PROCEDURE UPDATE\_SALEPRICE;

10

11 CALL UPDATE\_SALEPRICE;

Clear

Format

Get auto-saved query

☐ Bind parameters

[ Delimiter

;

]

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

Hide query box

Error

SQL query: [Copy](#)

DROP PROCEDURE UPDATE\_SALEPRICE

MySQL said:

#1305 - PROCEDURE Mysql\_learners.UPDATE\_SALEPRICE does not exist

**Congratulations! You have completed this lab on creating stored procedures in MySQL, and are ready for the next topic.**

**Author(s)**

[Lakshmi Holla](#)

[Malika Singla](#)

## Changelog

Date	Version	Changed by	Change Description
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 2021. All rights reserved.