



# INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

INGENIERÍA EN SISTEMAS COMPUTACIONALES

5to Semestre

Fecha de entrega: 11/12/2020

Actividad 2: Ejercicios STORED PROCEDURES & FUNCTIONS

Tema 5: SQL Procedural.

Materia: Taller de Base de Datos.

Nombre del Alumno: Marín Ramírez Mario.

Número de Control: S18070186

Correo electrónico: [mariomarin502t@gmail.com](mailto:mariomarin502t@gmail.com)

Profesor: I.S.C. Salvador Acevedo Sandoval.

Link tutorial: <https://www.mysqltutorial.org/mysql-stored-procedure-tutorial.aspx>

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'classicmodels' database selected, with 'Stored Procedures' expanded and 'GetAllProducts' highlighted. The main editor shows the SQL code for creating the procedure:

```

1 use classicmodels;
2
3 DELIMITER //
4 CREATE PROCEDURE GetAllProducts()
5 BEGIN
6     SELECT * FROM Products;
7 END //

```

Below the editor, the 'Result Grid' shows the output of the procedure, displaying a list of products with columns: productCode, productName, productLine, productScale, productVendor, and productDescription.

productCode	productName	productLine	productScale	productVendor	productDescription
S10_1678	1969 Harley Davidson Ultimate Chopper	Motorcycles	1:10	Min Lin Diecast	This replica features working kickstand, f
S10_1949	1952 Alpine Renault 1300	Classic Cars	1:10	Classic Metal Creations	Turnable front wheels; steering function; d
S10_2016	1996 Moto Guzzi 1100i	Motorcycles	1:10	Highway 66 Mini Classics	Official Moto Guzzi logos and insignias; sad
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Motorcycles	1:10	Red Start Diecast	Model features, official Harley Davidson log

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'classicmodels' database selected, with 'Stored Procedures' expanded and 'GetOfficeByCountry' highlighted. The main editor shows the SQL code for creating the procedure:

```

1 use classicmodels;
2
3 DELIMITER //
4 CREATE PROCEDURE GetOfficeByCountry(
5     IN countryName VARCHAR(255)
6 )
7 BEGIN
8     SELECT *
9     FROM offices
10    WHERE country = countryName;
11 END //

```

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'classicmodels' database selected, with 'Stored Procedures' expanded and 'GetCustomerShipping' highlighted. The main editor shows the SQL code for creating the procedure:

```

3 DELIMITER $$
4 CREATE PROCEDURE GetCustomerShipping(
5     IN pCustomerNumber INT,
6     OUT pShipping VARCHAR(50)
7 )
8 BEGIN
9     DECLARE customerCountry VARCHAR(100);
10
11     SELECT
12         country
13     INTO customerCountry FROM
14         customers
15     WHERE
16         customerNumber = pCustomerNumber;
17
18     CASE customerCountry
19     WHEN 'USA' THEN
20         SET pShipping = '2-day Shipping';
21     WHEN 'Canada' THEN
22         SET pShipping = '3-day Shipping';
23     ELSE
24         SET pShipping = '5-day Shipping';
25     END CASE;
26 END $$

```

Link Tutorial: <https://www.c-sharpcorner.com/blogs/store-procedure-with-real-time-scenario-in-sql-server>

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\marin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.19 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database scenario;
Query OK, 1 row affected (0.01 sec)

mysql> use scenario;
Database changed
mysql> create table product(ProductId int primary key, ProductName varchar(20) unique, ProductQty int, ProductPrice float);
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT Product VALUES(1,'Printer',10,4500);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT Product VALUES(2,'Scanner',15,3500);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT Product VALUES(3,'Mouse',45,500);
Query OK, 1 row affected (0.01 sec)

```

```

mysql> select * from product;
+-----+-----+-----+-----+
| ProductId | ProductName | ProductQty | ProductPrice |
+-----+-----+-----+-----+
| 1 | Printer | 10 | 4500 |
| 2 | Scanner | 15 | 3500 |
| 3 | Mouse | 45 | 500 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> Delimiter $$
mysql> CREATE PROCEDURE prcInsert(
-> in id int,
-> in name varchar(40),
-> in qty int,
-> in price float)
-> begin
-> DECLARE cnt int default 0;
-> DECLARE p float default 0;
-> select cnt=COUNT(ProductId) from Product where pname= name;
-> if cnt>0 THEN
-> update Product set ProductQty = ProductQty + qty where ProductName=name;
-> select p=ProductPrice from Product where ProductName=name;
-> if p<price THEN
-> update Product set ProductPrice=price where ProductName=name;
-> end IF;
-> else
-> insert Product values(id,name,qty,price);
-> end if;
-> end$$
Query OK, 0 rows affected (0.02 sec)

```

El siguiente link: <https://www.javatpoint.com/mysql-transaction> ya lo hemos realizado en la actividad 2 del tema 4.