Using Stored Procedures

(**Programming tip:** drop stored procedure if it exists, before and after creating each stored procedure, in order to release server resources.)

Log in to *YOUR* database using MySQL Workbench, AND through SSH.

*****First: drop *ALL* existing tables!*** Then…**

source db/gmc.sql

--or...

\. db/gmc.sql

List All Stored Procedures in MySQL:

```
SHOW PROCEDURE STATUS;
```

List All Stored Procedures in a specific database:

```
SELECT ROUTINE_TYPE, ROUTINE_NAME
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE ROUTINE_SCHEMA='database_name';
```

SHOW CREATE PROCEDURE Syntax:

```
SHOW CREATE PROCEDURE proc_name;
```

Reports:

Note: All currency values should be formatted with $ sign, including 2 decimal places. All zip codes and phone numbers should include hyphens. **(Sale's reps receive 3% commission.)**

1. Create a stored procedure that displays all vehicle information, in descending order of vehicle price (name it **sp_veh_info**).

2. Create a stored procedure that displays the total sales dollar amount for all sales reps (name it **sp_srp_total_sales**):

3. Create a stored procedure that displays the dealership ID and phone number, customer name, and the total amount spent on vehicles by each customer, sort by total amount in descending order (name it **sp_cus_total_sales**).

4. Create a stored procedure with one IN parameter (cityName) that displays all dealerships' names and phone numbers in a particular city, call it using the name of the city as the argument (name it **getDealerByCity**): stored procedures with one IN parameter (modification inside stored procedure does not change calling environment)

5. Create a stored procedure that modifies vehicle data, indicating that Steve Rogers purchased the vehicle with vin number 4S4BP67C454326621, use two parameters (p_cus_id, p_veh_vin). Modify all associated tables to reflect the purchase. Display before/after tables values w/in stored procedure (name it **setVehPurchase**)

6. Create a stored procedure with the necessary IN parameters in order to add another vehicle record, include a statement w/in the stored procedure that displays the added record (name it **addVehicle**). Call it passing the following values to each respective attribute:

```
vin=XKA478654WDBCA45E
dlr_id=2
asn_id=8
veh_type=Truck
veh_make=Chevrolet
veh_model=Silverado 2500HD
veh_year=2012
veh_cost=32000.00
veh_price=35000.00
veh_discount=0.00
veh_comm=0.03
veh_inventory_date=9/5/2011
veh_sold_date=today's date
veh_notes=Best in ratings!
```

7. Use the same stored procedure, however, initialize session variables to be passed as arguments to the stored procedure's parameters, then call it using these variables:

8. Create a stored procedure with one OUT parameter (any value passed to parameter is ignored by the procedure, and its initial value within the procedure is NULL) that counts the total number of vehicles. Before calling the procedure, initialize a session variable to be passed as an argument to the stored procedure's OUT parameter when called. Display variable's value before and after call. (name it **totalNumVehicles**):

9. Create a stored procedure with one IN parameter and one OUT parameter that gets the total number of customers from a particular state. Pass the state's name to the IN parameter. Also, before calling the procedure, initialize a session variable to be passed as an argument to the stored procedure's OUT parameter when called. Use the OUT parameter to assign the value of the total number of customers, based upon the state name, passed to the IN parameter, (name it **totalNumCustomers**):

10. Create a stored procedure (name it **spArea**) with one IN parameter (radius) and one INOUT parameter (area) that calculates the square foot area of a circular rug with a radius of 2 ft would take in a living room. Also, create a local procedure variable named pi. Return the calculated value back to the calling environment using the INOUT parameter:
    http://www.math.com/tables/geometry/areas.htm

Note (from MySQL Certification Study Guide): difference between OUT and INOUT

OUT: Any value the parameter has when it is passed is ignored by the procedure, and its initial value within the procedure is NULL.

INOUT: The value passed by the caller is the parameter's initial value within the procedure.