Due Date: Tuesday April 29, 2014 11:00 PM
Points: 35 points max
Turn In: The script and spool files turned in via the assignment drop box

General Directions

These tasks focus on the use of programming techniques to create user defined functions. For this assignment you will create two MySQL functions and use them in queries. You need to create and test the functions first, using a variety of values for testing and correcting your functions as needed.

After you have created and tested your functions, you set up a script to show me the source code for these functions and the test queries. Follow the script template for this shown at the end of this assignment.

Use the two part names for all tables. Create the tables in the a_bkinfo database.

Tasks: Functions to be created

You should test for and handle null parameters as explained in the function directions. You do not need to try to handle situations where the user of the function tries to pass in an argument of the wrong data type. A function will not execute if the arguments passed in do not match the data types of the parameters.

Create a function named **BookSize**. The function has one input parameter which is expected to be the number of pages in a book. The function returns a **string** that indicates how long the book is.

If the book has 300 pages or fewer, it is classified as a "Short" book.
If it has 301- 800, pages it is classified as a "Medium" book.
If it has 801- 1200, pages it is classified as a "Long" book.
If it has more than 1200, pages it is classified as a "ExtraLong" book.

If the input argument is null or a negative number, then the function returns the string message "InvalidInput".

Use the **If selection structure** in this function (not a Case expression).

Create a function named **PrevMonth**. The function has two input parameters (`in_date` and `in_mn_adjust`). The first is a date and the second is an integer which is expected to be the number of months to adjust. The function returns a string with the format 'YYYY-MM' which is the year and month for the month that is `in_mn_adjust` previous to the first parameter. See the sample return values below.
If the first parameter is null, use the current date as the first argument value.
If the second parameter is null or a negative number, return a null string.

For example,

```
a_bkinfo.PrevMonth('2014-04-10',  2) returns '2014-02'
a_bkinfo.PrevMonth('2014-04-10',  6) returns '2013-10'
a_bkinfo.PrevMonth('2012-04-10', 18) returns '2010-10'
a_bkinfo.PrevMonth('2014-04-10', null) returns null
a_bkinfo.PrevMonth(null, 0) returns '2014-04' -- if run in April 2014.
```

Tasks: Function Demonstrations

We want to demonstrate that the function works by supplying a variety of arguments. For task 1, you are to use the technique described in the notes for this unit to set up a virtual test table for the function. For other tasks you use the function with the database tables.

Task 01: **BookSize:** Demonstrate your function by running a query using a virtual test table to supply arguments. Include enough rows to fully demonstrate that your function is correct. The result of running this query should follow this sample run- I have shown only some of the rows.

```
Sample rows only
+---------+-----------+--------------+--------------+
| testrun | pagecount | actual       | expected     |
+---------+-----------+--------------+--------------+
|       1 |      NULL | InvalidInput | InvalidInput |
|       5 |       250 | Short        | Short        |
|      11 |      1199 | Long         | Long         |
|      12 |      1200 | Long         | Long         |
|      14 |      1500 | ExtraLong    | ExtraLong    |
+---------+-----------+--------------+--------------+
```

Task 02: Use the **BookSize** function to produce a display as shown here**.** Use the books table as the data source for this query. The order of the rows must match the order shown here. Do not include any books with a null or negative page count.

```
Sample rows only
+-----------+------------+
| Size      | NmbrBoooks |
+-----------+------------+
| Short     |        234 |
| Medium    |         87 |
| Long      |        201 |
| ExtraLong |          3 |
+-----------+------------+
```

It is possible that the books table might not have any books of one of these size categories; in that case your result set might not have a row for that size category.
Optional (but not extra credit) Display a row for each of the size categories, even if we have no books with a page count for that category; display a count of 0 for that row. (Hint: consider a virtual table of size categories and Inline views.)

For tasks 03 and 04, careful use of the function will save you a lot of work.

Task 03: Use the **PrevMonth** function to display customer id and name for all customers who have at least one purchase in every one of the three months as defined in assignment 10 - a three month period starting four months ago and extending for two months. You need to derive the months based on the system date. Use the tables in the books databases as the data source for this query.

```
Sample rows only
+-------------+----------------------+
| customer id | customer name        |
+-------------+----------------------+
|      227105 | Kafka, Franz         |
+-------------+----------------------+
|      261502 | Hawthorne, Nathanile |
+-------------+----------------------+
```

Task 04: Use the **PrevMonth** function to display the number of orders we had in the previous two months and the number of customers we have who have at least one order in the previous two months. The term "previous two months" means any date in the two month before the current

month. So if you run the query in April 2014, the query will return data for Feb 2014 and March 2014. You need to derive the months based on the system date. Use the tables in the books databases as the data source for this query.

```
+--------------+----------------------+
| NumberOrders | NumberCustWithOrders |
+--------------+----------------------+
|          453 |                  101 |
+--------------+----------------------+
```

Template for the Script file

This is similar to the regular template but includes two statements that use show create function to show me the source code for your functions.

```
comment with your name
\W

use a_bkinfo;

/*  TASK 00 */
select user(), current_date(), version(), @@sql_mode\G

/* function source code */
show create function  BookSize\G

/*  TASK 01 */
replace this line with your SQL for task 01

/*  TASK 02 */
replace this line with your SQL for task 02

/* function source code */
show create function  PrevMonth\G

/*  TASK 03 */
replace this line with your SQL for task 03

/*  TASK 04 */
replace this line with your SQL for task 04
```