Due Date: Thursday 2013-07-04 11:00 p.m.
Points: 40 points max
Turn In: The zipped file containing the script and spool files.

General Directions

Use the books tables in the databases a_bkinfo and a_bkorders. The script for these databases is on the Download page. **Use the two part table names for all queries. This is important so that tables from either database can be accessed**.

Since these tables have a lot of rows, the tasks will often limit the number of rows to display. You might wish to test run the query without the limit which can help find errors.

Tasks

Task 01: Use the Find_In_Set function to display the customer id, customer last name and the book title for all books ordereds but exclude the orders by customers in the Midwestern states.
Order the result set by the book title and limit to 20 rows.

The Midwestern states are defined as: Illinois, Indiana, Iowa, Kansas, Michigan, Minnesota, Missouri, Nebraska, North Dakota, Ohio, South Dakota and Wisconsin. (I am sure you can find the state abbreviations for these.

Task 02: We are concerned that some books might not have the correct topic id values. Display the book id and title for any book that contains the string "SQL" in the title but the book is not classified as an SQL topic book. Be certain to use the topic id 'SQL' and not the topic description for this task.
You may use a regular expression for this or the strings functions; do not use a wildcard pattern with the Like operator. We are not testing for the word 'SQL', a book with the title MySQL contains the string "SQL" in the title.
Suggestion: use a variable for the topic_id and test with different values for the topic id.

Task 03: Display the book id, the first 20 characters of the title, list price and the list price rounded to the nearest $10 for the books with a topic of poetry (POE) or programming (PGM). The rows are to be ordered by the rounded price; use the book id as a tie breaker for the sort. If a book has a topic for both POE and PGM then display the book only once.

Task 04: Display the customer id, the order id and the order date for all orders that were placed on the first day of any month.

Task 05: Display the customer id, the order id and the order date for all orders that were placed on the last day of any month.

Task 06: Display the customer id, the order date and the age of each order.
The age of an order is determined by the number of days between the current date and the order date.
Sort the data by the customer id and order date. Limit the display to 20 rows.
The sample display shows order age if the query were run on July 01, 2013. You will run this using the current date. Using a variable for the date to be tested makes it easier to test this with different date values.
Do not include orders with a future date.

```
Sample output
+---------+------------+--------------------+
| cust_ID | order_date | orderAged          |
+---------+------------+--------------------+
|       1 | 2013-07-01 | less than 14 days  |
```

```
|       2 | 2013-06-15 | less than 30 days  |
|       3 | 2012-06-22 | 180 days or more   |
```

The values used for orderAge are

- Less than 14 days
- Less than 30 days
- Less than 90 days
- Less than 180 days
- 180 days or more

Task 07: Display the customer id, last name and the date their account was opened for customers who opened their account during the previous 8 months. This query should run correctly no matter when it is run. For this query **a "previous" month is defined as the entire month before the current month.** If the current month were Oct 2012, the previous month would be Sep 2012 and the two previous months would be Aug 2012 and Sep 2012. In this case, a date in October 2012 would not be in the previous month.

Task 08: We are going to calculate a discount for orders. The discount rate will be a random value between 1% and 5%. Use the rand function to generate a discount rate in the range 0.01 through 0.05 and round to two digits after the decimal. Apply that calculated value to the extended cost to get the discount amount
Each row can have a different random discount percent applied.
Display the order id, book id. title, the extended cost and the discount amount.

```
Sample rows- here the first two rows show got a 1% discount and the third a 5%
discount
+----------+---------+-------------------+----------+-----------------+
| Order ID | Book id | Title             | Ext Cost | Discount Amount |
+----------+---------+-------------------+----------+-----------------+
|      103 |    1004 | Summer Fun- When? |   310.00 |            3.10 |
|      103 |    1008 | Getting Real      |   110.00 |            1.10 |
|      104 |    1008 | Getting Real      |   310.00 |           15.50 |
```

THE END