Due Date: Saturday, October 26, 2013 11:00 PM
Points: 40 points max
Turn In: The script and spool files turned in via the assignment drop box

General Directions

Use the two books databases: a_bkinfo and a_bkorders.

The general directions for this assignment are critical; this is an assignment where people can do a lot of work and get no points because they do not follow the directions.

This assignment focuses on the use of the Set operators. Since MySQL does not support all the set operations directly, you will need to use subqueries to implement most of these tasks. MySQL does support the Union operator so use that when possible for the task logic.

I have provided a view (a_bkorders.bkv_CustWithOrders) in the books sql script that joins tables that you should use as the table expression in the From clause. In the sql that you write for the tasks, you are not allowed to use an inner join, outer join or a comma join. (The view contains joins- that is not a problem- you do not write joins in your queries.) You will need to use subqueries frequently.

Do not use the count function.

Definition of terms:

We want to look at some patterns of customer buying habits for specific months

We are concerned about book sales to customers in a three month period starting 6 months ago and extending for two months. For example:

If the current month is	We want sales from the months
November 2012	May 2012, June 2012, July 2012
April 2012	October 2011, November 2011, December 2011
Jan 2013	July 2012, August 2012, September 2012

These three months are referred to as "the months under discussion". The oldest month is referred to as the 'first month". So if I run this in April 2012, the 'first month' is October 2011; the 'second month" is November 2011and the "third month" is December 2011.

The buying patterns we are looking for are of the nature of who bought books in each of the three months under discussion, who bought books in at least one of those months; who bought books in the first two months but not the third. We want information about customers who bought books and the order dates. Since we will use this result set in several tasks, you use the provided view for the queries.

You need to set up variables that will be used to test for the required three months (This is done in Task 1.) You have to calculate these months based on the run date of your script. You are required to use these variables in the rest of the queries for the filters.

See also the note at the end of the assignment.

I am not saying exactly how many variables you need to use or the nature of those variables. The more precise your variables are defined in the first step, the easier it will be to write the queries for the rest of the assignment. Some people use a minimum number of variables and repeat calculations in each query; that is harder to read and maintain than putting more of the calculations into the definition of the variables and having simple queries.

It is possible that some of the queries will return no data.

The output for tasks 3- 10 will all have this format. These should not have any duplicate rows.

```
+--------+----------------------+
| c_id   | c_name               |
+--------+----------------------+
| 211483 | Carroll, Lewis       |
| 217796 | Anders               |
```

Tasks

Task 01: Set up variables to calculate the three months based on the current date as described in the general directions.

Start with one variable named @rundate. For the script you turn in, you will assign the current date to this variable. set @rundate := current_date;
But for checking if you are getting the correct dates for testing, you could assign other values to this variable and compare them to the table above.
Remember to reset the variable to current_date in your final version of the script.
I need to see the calculations you do for each variable.
Display the values of your variables using a single Select statement. If you have more than 4 variables use a \G as a query terminator.

The next task displays data for each month. This is to help you see if you have set up the variables correctly. You can use the results of these queries to help you analyze the results you get from the rest of the queries. You cannot prove a query is correct by inspecting the output- but you should be able to detect that some queries are not providing the correct results.

Task 02: This task has three separate selects. For the queries in this task, use the view as the data source and display all of the columns returned by the view. Use your variables to show the information about customers and books they bought during they bought during the **first** month of these three months. Then use your variables to show the customers and books they bought during the **second** month of these three months. Then use your variables to show the customers and books they bought during the **third** month of these three months.

The output for tasks 3- 10 will all have this format. These should not have any duplicate rows.

```
+--------+----------------------+
| c_id   | c_name               |
+--------+----------------------+
| 211483 | Carroll, Lewis       |
| 217796 | Anders               |
```

Task 03: Display customer id and name for all customers who have at least one purchase in **at least one** of the three months under discussion. Do this as a Union query.

Task 04: Display customer id and name for all customers who have **no** purchases in **any of the three months** under discussion. Since you are using the view as the data source these customers will have some orders but not in the three months under discussion.

Task 05: Display customer id and name for all customers who have at least one purchase in **each** of the three months under discussion.

Task 06: Display customer id and name for all customers who have at least one purchase in **the first month but not in the second and not in the third** of the three months under discussion.

Task 07: Display customer id and name for all customers who have at least one purchase in **the first of the three months and at least one order in either or both the second or in the third** of the three months under discussion.

Task 08: Display customer id and name for all customers who have at least one purchase in **the first of the three months and at least one order in either second or in the third (but NOT both)** of the three months under discussion.

Task 09: Display customer id and name for all customers who have at least one purchase in **either or both of the first two of the three months but not in the third** of the three months under discussion.

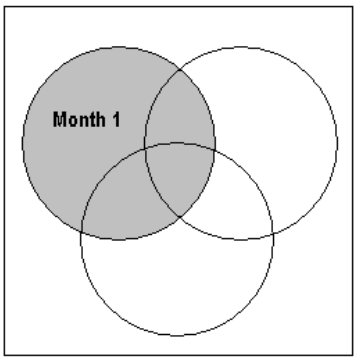
Task 10: Display customer id and name for all customers who have purchases in **exactly one of the months** of the three months under discussion. For example, the customer might have a purchase in the second month but not in the first or third month.

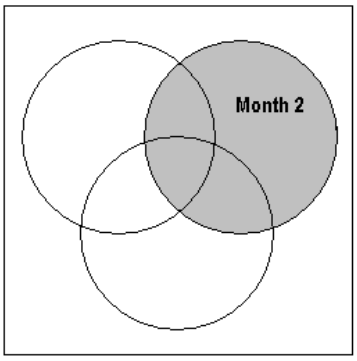
Note: sometimes people cannot get the date calculations correct for the variables. That would create a problem for them with the rest of the assignment and I do not want to penalize you for all assignment tasks. If that is your situation then you should hard code in the data into the variables to use the following three months: April 2013, May 2013, and June 2013. This will give you three months to use for the queries. You will lose 10 points on the assignment but you should be able to work with the rest of the assignment.

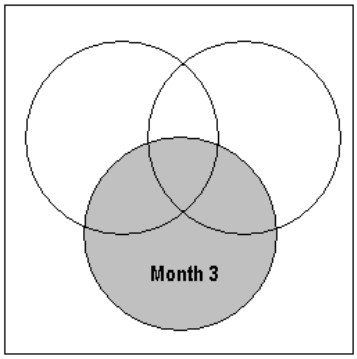
Some people find Venn diagram helpful for this topic- and some people don't. If this does not help, then don't worry about the diagrams.

These are Venn diagrams showing sales for Month1, for Month 2 and for Month 3

The rectangle represents all sales and the shaded circle represents one of the month's sales.







These diagrams represent other possible combinations of sales by month. Try to figure out what each of the diagrams represents- some of these correspond to tasks in the assignment.