Due Date: Nov 11, 2015 11:00 PM (Late date Nov 12, 11:00 pm) Points: 35 point max

General Directions

Use the books database tables.
These tasks focus on the use of Subqueries. Consequently, you must use subqueries to solve the problems. In many cases you could solve the task without the use of subqueries- but that will not earn any credit for the assignment. You may not use the set operators for this assignment.

Follow the same list of general rules as for Assignment 11 with one exception. **You may use correlated subqueries**; with a correlated subquery there will be a join between a table in a subquery and a table in an outer query with the join condition being expressed in the Where clause. This does not mean that you can use joins in general- only the join used to create a correlated subquery. You should probably review the document in the previous unit on correlated subqueries.

You are not required to use correlated subqueries if you find another way to do the task.

You are not allowed to use any views.

The tasks which require an Exists operator can also use a Not Exists operator. If the task says to use an Exists query and you do not use Exists in a meaningful way, you will get no points for that task.

These queries will not have duplicate rows in the result set.

Do not hard code a literal for the current year.

Each query that uses a join - other than the correlation join- will lose 7 points.

Tasks

Task 01: For each customer in the customers table who lives in New Jersey (NJ) or in Massachusetts (MA), display their id, their last name and the number of orders they have. Sort by the customer id.

```
Cust_ID Cust_Name_Last       Number Of Orders
------- -------------------- ----------------
300124  Adams                              21
300150  Jobs                                1
300153  Jones                             123
```

Task 02: Use an Exists query to display the customer id and last name for any customer who ordered a book in each (every one) of the first three months of the current year. Sort by the customer id. (The first three months of any year are always Jan, Feb, Mar- so you can hard code the months.)

Task 03: Use an Exists query to display the author information for authors who have more than one book but for whom we have no book sales. Sort by the author id.

```
Author_Name_First    Author_Name_Last     Author_ID
-------------------- -------------------- ---------
Mark                 Gersten              G1234
Sue                  Haldeson             H9876
```

Task 04: We want to find any books which cover **exactly** two of the different sql systems using the topic ids 'SSRV', 'ORA', 'MYSQL'
For example, the book could have a topic of SSRV and a topic of ORA, but not MYSQL. Any two of these topics is sufficient to pass our filter. Display the id and title of the book(s) that are meet this test.

Task 05: For each order placed in the first three months of the previous year, display the following pieces of data:
order date
order id
customer id
customer last name - use an alias of customer
total number of books purchased (quantity) on that order- use an alias of NumberBooks
total amount due for that order - use an alias of OrderCost.
Sort the result by the order date.(Suggestion- use the order headers table in the From clause of the main query.) Remember you are not allowed to use joins other than that required for a correlated subquery.

Task 06: Display the customer id and last name for customers with at least three and no more than five orders last year. Sort by the customer id.

Task 07: Display the cust id and last name of customers who had the same or more orders in the first quarter of last year than they had in the first quarter of the current year. Count the order id values for this comparison. This will include only customers who have any orders in those two time periods. Sort by the customer id.