Due Date: Sunday 2013-07-14 11:00 p.m.
Points: 45 points max
Turn In: The zipped file containing the script and spool files.

General Directions

Use the books databases.

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. Many of these will use aggregate functions.

- You must do each task with a single query. The query will have sub query components but it will be a single query.
- Do not use a variable.
- Do not use a join of any kind except you may use a correlated subquery for the tasks which specifically allow this. (Tasks 6,7,8,9)
- Do not use the Union operator
- If you use column aliases, then use different column aliases in each subquery. Using the same column alias may be legit but it makes your query very hard to read.
- A book with an order quantity of 0 is still considered a book that is ordered. (Some people are testing that the quantity > 0- do not do that to determine if a book has been ordered.). An order header without any detail lines is still an order.

Queries that use sub queries tend to be longer and harder to read if they are not formatted properly. Queries that are hard to read will lose points.

For the tasks that allow a correlated subquery: 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.

Tasks

Task 01: Display the book id and title for any books which someone has ordered and the book is categorized as **both** an SQL book and a database book. Use the Topic_id to filter for DB and SQL. Sort by the book_id.

Task 02: Display the ID and title of the books with the largest number of sales; include ties. For this query, use the total quantity sold when determining the sales of a book.

Task 03: Display the customer id and last name for customers with no more than 5 orders. Do not include any customer with no orders. Sort by the customer id.

Task 04: For each book in the books table that includes 'Bird 'in the book title, display the book ID and title and a message as to whether or not we have any orders for that book. Do not use the Count() function.
The output will follow this format and be sorted by the OrderStatus column with the books with no order first; the second sort key is the book id.

```
Sample rows only
+---------+-----------------------------------------------+-------------+
| book_id | title                                         | OrderStatus |
```

```
+---------+-----------------------------------------------+-------------+
|    2029 | The Forgotten Bird Strikes Back               | No Orders   |
|    1104 | Sibley Guide to Bird Life and Behavior        | No Orders   |
|    1543 | Birding and Implementing a Birdhouse          | Have orders |
|    1085 | My Bird Flicker                               | Have orders |
```

Task 05: 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 06: You may use a correlated subquery for this task.
Use an Exists query to display the customer id and last name for any customer who ordered a book in each of the last three months of the previous year. Sort by the customer id.

Task 07: You may use a correlated subquery for this task.
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. Some books have more than one author; for this task we do not care about the sequence number for the author.

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

Task 08: You may use a correlated subquery for this task.
Display the customer id and last name for each customer in the customer table and the date of their earliest order and the date of their most recent order, the number of orders they have and the total sales for all of their orders. If the customer has no orders, display the message "No Orders" in the last 4 columns. Use sensible column aliases for all columns.

Task 09: You may use a correlated subquery for this task.
You many have noticed data in the orders table such as the rows below where we have one order (order_id 33034) where the same book (book_id 1619) occurs more than once. This is allowed because the pk for this table is (order_id,order_line) .

```
order_id  order_line  book_id  quantity  order_price
--------- ----------- -------- --------- -----------
33034     1           1619     1         29.99
33034     2           1619     3         15.95
```

Write the query to display the customer id and last name for any customer who ordered the same book more than once on an order.
Note that the query displays cust id and name; the query does not display the order details.

THE END