# CS 155A | Fall 2013 Assignment 11

Due Date: Saturday, November 09, 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 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.
Follow the same list of general rules as for Assignment 10 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 are not required to use correlated subqueries if you find another way to do the task.
Do not hard code a literal for the current year.
A query that uses a join - other than the correlation join- will get no points. A From clause will reference only one table.
If the query says to use a specific technique, you need to use that technique in a meaningful way to get any points for the task.

## Tasks

**Task 01:** For customers in the customers table with a cust_id of 250000 or less, display their id, last name and the number of orders they have. Sort by the customer id.

```
+---------+----------------+------------------+
| cust_id | cust_name_last | number of orders |
+---------+----------------+------------------+
|  200368 | Blake          |               45 |
|  202958 | Denver         |                0 |
|  208950 | Adams          |              102 |
```

**Task 02:** 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 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     |
| Susan             | Haldeson         | H2234     |
```

**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 meet this test. Remember you may not use the set operators.

**Task 05:** The book authors table includes a column named author_sequence; this sequence affects the order in which the authors are listed when a book has multiple authors.
We want to know if we have any authors who are listed as the first author (author_sequence of

1) on some book and also are listed as something other than the first author on another book.
For example, suppose Sam Jones (author id J345) is listed as author 1 on "Applied Zen" and as author 2 on " Birds and Cats Together "
For each of these authors, display the author name, book title and author sequence. Do not make any assumptions about how many authors are allowed for a book.
Suggestion: use the book authors table as the table in the main From clause and pick up the other data as subqueries.

```
+------------------+----------------------------------+-----------------+
| Author           | Title                            | author_sequence |
-------------------+----------------------------------+-----------------+
| Jones, Sam       | Applied Zen                      |               1 |
| Jones, Sam       | Birds and Cats Together          |               2 |
```

**Task 06:** For each order placed in the second quarter of the current 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 ( use the quantity) on that order- use an alias of NumberBooks
total amount due for that order - use an alias of OrderCost

**Task 07:** 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)
This is possible table data

```
+----------+------------+---------+----------+-------------+
| order_id | order_line | book_id | quantity | order_price |
+----------+------------+---------+----------+-------------+
|    33034 |          1 |    1619 |        1 |       29.99 |
|    33034 |          2 |    6789 |        1 |       25.95 |
|    33034 |          3 |    1619 |        5 |       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.
The query displays cust id and last name; the query does not display the order details.
Sample Run

```
+---------+----------------+
| cust_id | cust_name_last |
+---------+----------------+
|  332447 | Wilson         |
```