

Due Date: Nov 25, 2015 11:00 PM (Late date Nov 26, 11:00 pm)

Points: 35 point max

## General Directions

This assignment uses the tables associated with the books database.

These tasks focus on the use of Advanced Queries programming techniques. These queries must use the techniques described in the notes for this unit.

Review of basic concepts re sales and grouping

- Some of these tasks refer to the list price of the book- that is the attribute in the books table
- Other tasks refer to the "sales of books"; to get sales values you need to work with books that have been ordered and for that you need the order details table and possibly other tables . Do not include books with no sales.
- If the query determines total quantity- that is total number of books ordered for whatever grouping is required.
- If you want to display data by book, you need to group by the book id. You might add additional attributes to the Group By clause for which there is only one value per book, such as title. But if you add the order\_id attribute to the Group By clause, then you are no longer displaying data by book; you are instead creating one group for each book and order id combination.
- These queries may produce a lot of rows. Do not limit the rows unless the task specifies that.
- Sorting the result set is important for this assignment.
- Do not create views for this assignment; you can put a subquery in the From clause instead or use a CTE.
- Alignment of descriptive labels in the columns is not critical.
- Some of these queries are quite long. If you do not format the sql appropriately and indent subqueries, then you may lose points if I cannot quickly read your sql logic.

For many of these tasks, using one or more CTEs can be helpful in separating the steps to accumulate the data; the steps to rank, partition or otherwise analyze the data; and the steps to format the data. Do the formatting in the last step since that will commonly mean that a numeric sales value is formatted using To\_Char which produces a string. You do not want to do numeric calculations with strings.

The column command shown here will limit the width of a column with an alias of "Book\_Title" to 25 characters, truncating any extra characters. You can include this in your script to make the formatting of that column easier. I have used this for the sample displays. You can add other column commands to limit the width of character columns. Do the formatting of the numeric data ( such as price, total sales) and dates in the result set using to\_char.

Follow the sample display for the format, column aliases..

```
Column "Book_Title" format A25 truncated
```

## Tasks

**Task 01:** Display the following columns for each book in the books table.

column 1: the row\_number using the list price as the order field. This uses the row\_number function and a windowing clause.

column 2: book id

column 3: the title

column 4: publisher id

column 5: list price.

Sort the output by the list\_price.

ListPrice	Order	Book ID	Book_Title	Publ	Price
	1	100	Queires Are Fun	8000	12.00
	2	560	The Subquery Model	8001	15.25
	3	1874	Even More Subqueries	8009	17.95

**Task 02:** Display the following columns for books with a publication date more recent than 2005.

column 1: a ranked ordering by the book's publisher and the list price with the more expensive books first within the publisher id. The first column reports any tied rows with the same price as having the same rank number.

column 2: publisher id

column 3: list price

column 4: book id

column 5: the title.

For example, all books from the same publisher that cost \$35.00 will have the same number. But another book from a different publisher that costs \$35.00 will have a different number. Do not skip any numbers in the first column.

Sort the final output by publisher id and the list price desc

Publ-ListPrice	Order	Publ	Price	Book ID	Book_Title
. . MID WAY THROUGH THE OUTPUT					
	16	8223	59.95	456	Wait for June
	17	8223	35.00	789	Yet Another Subquery
	17	8223	35.00	874	The Exam Cometh
	17	8223	35.00	451	The SQL Booklet
	18	8223	25.00	652	The Big Book of SQL
	19	9358	59.95	800	Lichens
	20	9358	35.00	900	Moss Gardening

**Task 03:** Which are the top 15 selling books based on the total quantity ordered for the book? Round the total sales to the nearest ten before ranking the books.

Display the Sales Ranking number, book id, title, and total quantity sales in sales ranking order. Please pay attention to the values in the first column of the sample display for ties and follow that ranking pattern.

All books that are tied for the last place should be returned. The only place we are concerned about ties is in the last slot.

Although it is not required, consider using a CTE to encapsulate the table joining syntax.

Sales Ranking	Book ID	Book_Title	TotQuantSale
1	1245	The Ants	5,100
1	1784	The Beatles of Bolivia	5,100
3	9845	Mosses of Pennsylvania	1,830
4	7638	Guide to Tree Bark	1,720
5	9298	Why Ducks Matter	1,500
. . . rows omitted to save space here			
15	8178	SQL Legends	500
15	8005	Logic Matters	500

15 rows selected

**Task 04:** Display a running total of the total sales by order date for each customer. Limit this to sales in the current year. If the customer has more than one order on the same day, these will be combined into a single row. Note that the total restarts for each customer.

To reduce the output, limit the customers displayed to those with a customer id less than 220000. Do that

filter in the main query ( i.e. not in a CTE or subquery).

Running totals are discussed in the notes. I highlighted some rows to emphasize the breaks in the customer id values. You are not expected to highlight any rows.

cust_id	order_date	SalesByDay	RunningSalesByCustomer
200368	2015-03-08	225.00	225.00
212921	2015-04-05	40.49	40.49
212921	2015-05-01	165.00	205.49
212921	2015-05-24	75.00	280.49
218709	2015-03-08	441.52	441.52
221297	2015-01-18	564.95	564.95
221297	2015-02-12	171.86	736.81

**Task 05:** Rollup total sales for each book by author.

The third column is the total quantity of books sold and the fourth is the total sales.

Include rows if an author or book title has no sales but do not include authors with no books.

Only use the sales data for the first author of a book- use the sequence attribute in the book\_author table. For this task you may assume that there will always be an author with sequence 1 for each book. This means that if a book has three authors, we include the sales data only under the first author and not with all three authors.

Note that there is a grand total line at the bottom.

To keep the output shorter limit this to authors with an ID that is less than 'J'. You can do that filter in the CTE.

The following is typical output; format your output to match this display.

Author ID	Book ID	Total Quantity	Total Sales
ABCDE	1948	6	125.64
ABCDE	Total	6	125.64
B3456	1142	19	506.80
B3456	Total	19	506.80
B8018	1102	2	100.00
B8018	1670	60	1200.00
B8018	1689	13	700.45
B8018	Total	75	2000.45
C5820	1899	0	No sales
C5820	Total	0	No sales
. . . . .			
GrandTotal	Total	3083	89372.42

**Task 06:** Calculate a 5 day **moving sum and moving average** for total sales of books; use current row and 2 days preceding and 2 days following. For example, the data for the 11th of the month would use the data for the 5 day span of the 9th, 10th, 11th, 12th, and 13th.

Limit the result set to orders placed in March 2014 and display a row for each day in March 2015 even if we have no sales that day. Display 0 for days where there are no sales. Before you go any further, think about how many days there are in the month of March.

The display has the layout shown below. Display the order date, total sales for that date, the 5 day sum and the 5 day average. The result is in order date order.

This sample output uses data that are not in your tables. Your report does not round sales.

Date	TotalSales	5 DaySUM	5 DayAvg
01-Mar-2015	400.00	860.00	286.67
02-Mar-2015	310.00	2,110.00	527.50
03-Mar-2015	150.00	3,110.00	622.00
04-Mar-2015	1,250.00	2,710.00	542.00
05-Mar-2015	1,000.00	2,400.00	480.00
06-Mar-2015	0.00	2,250.00	450.00
07-Mar-2015	0.00	1,000.00	200.00

. . . continue to the end of the month

This is a good place to use CTE to break this down into pieces. You will need to combine ideas from more than one demo for this task.

You may also want to use variables for various date components to make this more generic.