

Due Date: Sunday 2013-07-07 11:00 p.m.  
 Points: 40 points max  
 Turn In: The zipped file containing the script and spool files.

## General Directions

This assignment uses the tables from the a\_bkinfo and a\_bkorders databases. You need to spend some time reading the create table statements and looking at some of the data before you try writing these queries.

Many of the remaining assignments use these tables.

Use the two part table name in all queries.

When you are asked to find a book on a certain topic such as a database book, use the topic id- for example the topic id for database books is DB; the topic id for fiction books is FCT. A list of the topic ids and descriptions is in the topics table. If the query asks you to display the topic description, then you need to link to the topics table.

Several tasks ask you to filter for books in defined sets of topic id values called **categories**; use these lists for those tasks. Some topics are not included in a category; some topics are in more than one category.

Category	Topic id
Science	SCI
Programming	PGM, VB, NET, ADO
Database Systems	DB, SQL, SSRV, MySQL, ORA, ADO
Literature	FCT, POE

Things to think about as you do the assignment.

- What is the difference between Count (...) and Count (distinct ...)? If the attribute value you are counting can occur more than once, do you want to count each occurrence? How is this affected by joining several tables in the From clause?
- Do you need inner join or outer joins? ( Hint- you will not need a full join.) Do not use an outer join if an inner join is sufficient; an outer join requires more resources. Do not use an outer join and then throw away the rows you added due to the outer join.
- How do outer joins affect which column you use for the aggregates?
- Case expressions are useful for some situation where you need to do different things with different values. If your only concern is if something is null or not, then coalesce takes less typing.
- Ignoring the definition of the categories, given above, is not going to help your assignment score.

## Tasks

**Task 01:** Consider all books that have been ordered. Display the book id, the average price at which the book was sold and the total number of copies of the book ordered. Round the average price to the nearest dollar.

Book_id	AvgPrice	CopiesOrdered
1077	123	11
1101	55	854
1102	67	42
1103	8	185

**Task 02:** For each topic **that we have in the topic table**, display the topic id and description and the number of books we have for that topic id and the average list price for that topic. Include all topic areas from the topics table even if we have no books on that topic. The average list price is displayed **formatted** with two digits after the decimal point and the string 'n/a' is displayed if we have no books for that topic or if all the books we have for that topic are missing a value for list price. Order by the topic id.

ID	topic_descr	NumberOfBooks	AvgListPrice
ADO	ADO	2	54.99
ART	Arts, Photography	1	21.29
CMP	Computer Science	0	n/a
FCT	Fiction	3	29.95
HIST	History	1	n/a

**Task 03:** Display the number of orders we had in the previous month and the number of customers we have who had at least one order in the previous month. The term "previous month" means any date in the month before the current month. So if you run the query in Oct 2012, the query will return data for the entire month of Sep 2012. If you run the query in Mar 2013, the query will return data for the entire month of Feb 2013. The term "previous month" does *\*not\** mean the past 30 days. The output is one row.

NumberOrders	NumberCustWithOrders
47	17

**Task 04:** How many orders do we have which include an order for a science book? Use a variable for the topic id for this test. Display one column, one row- just the number of orders.

NumOrders Indicated Topic
123

**Task 05:** For each publisher that we have in the publisher table, list the publisher id and publisher name as the first column and the date of the most recent order for books from that publisher as the second column and the oldest order as the third column. If there are no orders for books from that publisher, display the message 'No Orders' instead of the date. **Format** the date as shown in the sample display.

Publisher	MostRecentOrder	OldestOrder
9000 Microsoft Press	2012-10-06	2012-10-06
9030 McGraw Hill	2013-04-08	2010-08-08
9102 Alfred A. Knopf	No Orders	No Orders
9325 Addison Wesley	2013-03-03	2011-10-28

**Task 06:** Display the id and name of all publishers who publish more than 3 but no more than 10 books in the Programming category.

**Task 07:** This is a cross tab query. We want to know how many books we carry in each of the indicated **categories**. Display the result as a single output row. The categories are:

- Science
- Programming
- Database Systems
- Literature

There is also a column for all books we have.

If a book has more than one topic id in a category, count it only once. If a book is in more than one category, then it counts in each of those categories. For example, suppose we have a book that has the following topic ids: VB, Net. This book is counted once in the Programming category. Suppose we have another book that has the following topic ids: FCT, POE, SCI, and SQL. This book is counted once in the Science category, once in the Literature category, and once in the Database Systems category.

Science	Programming	Database/SQL	Fiction/Poetry	all Books
12	22	37	7	74

**Task 08:** We want a display of the order date (year and month only) and the number of orders in each month and the total sales amount each month. The display will look like this. There is one row for each month that we have in our orders table. The first column is the year and month with the month shown as the three letter abbreviation. The second column is the number of orders that month. The third column is the total sales for that month. The display is sorted by the year and month in calendar order.

Year-month	NumberOrders	TotalSales
2012-Nov	7	8050.08
2012-Dec	4	30489.53
2013-Jan	9	16865.88
2013-Feb	8	14736.61
2013-Mar	4	3021.40
. . .		

THE END