Due Date: Tuesday May 6, 2014 11:10 p.m.
Points: 35 points max
Turn In: The script and spool files turned in via the assignment drop box

General Directions

Use the books databases. Use the full table name for the tables including the database name- such as a_bkorders.order_headers

These tasks focus on the use of Advanced Queries programming techniques. Each task uses 1 query. Do not create views; use subqueries. The notes for this week sometimes give you more than one way to accomplish a task; you can use any of these you wish. The data in the sample displays is to show aliases and format and layout only- it is not necessarily the correct answers. And the totals are not always numerically correct.

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
- Several questions in this assignment refer to total sales; this means that you multiply the quantity times the order price and total that value for the indicated group. **One more time:** The amount due on an order is not the sum of the prices; the amount due also reflects the quantity purchased. You need quantity * price. You will lose points on **each** query where you do not calculate this correctly. If you buy 25 books, it will cost more than if you buy 1 book. The price in the order details table is the price per book.
- 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 getting one group for each book and order id combination.
- These queries may produce many rows. Do not limit the rows.
- Sorting the result set is important for this assignment
- Do not create views or temporary tables for this assignment; you can put a subquery in the From clause instead
- 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 will less partial credit if I cannot quickly read your sql logic.

Tasks

Task 01: Display the following for book orders; each row shows data for a single order date. The second column is the number of orders on the given date; the third column is the amount due for those orders, the fourth column is the number of books ordered for those orders. The last row shows the grand totals. The result is sorted by the order_date.

```
+------------+-------------+----------+----------------+
| OrderDate  | NumberOrders | AmntDue  | NumbBooksPurch |
+------------+-------------+----------+----------------+
| 2012-06-12 |           34 |   123.45 |            323 |
| 2012-07-18 |           19 |   456.78 |             45 |
. . . rows omitted to save space here
| 2013-11-22 |            3 |    99.00 |              6 |
| NULL       |           76 | 85200.22 |            788 |
+------------+-------------+----------+----------------+
```

Task 02: Display the book order data showing the year of the order in the first column, the month in the second. Display totals by month and by year and grand totals. The result is sorted by the year

and month. Display the messages "Yearly Total" and "Grand Total" instead of the nulls that is the default display for these rows

Year	Month	Order_ID	AmntDue	NumbBooksPurch
. . . rows omitted to save space here				
2012	11	2005	34.00	3
2012	11	2016	78.00	4
2012	11	3005	90.90	2
2012	11	NULL	999.90	99
2012	12	895	150.00	20
2012	12	899	50.60	1
2012	12	NULL	999.90	99
2012	Yearly Total	NULL	44444.90	999
2012	1	1152	99.98	2
2012	1	1153	69.00	3
2012	1	1155	564.95	12
. . . rows omitted to save space here				
2012	Yearly Total	NULL	55555.09	888
Grand Total	Yearly Total	NULL	99999.99	1887

OPTIONAL For more of a challenge try this format;(If you do this use the comment Task 02 format 2)

Year	Month	Order_ID	AmntDue	NumbBooksPurch
. . . rows omitted to save space here				
2013	11	2016	78.00	4
2013	11	3005	90.90	2
2013	11	Month Total	999.90	99
2013	12	895	150.00	20
2013	12	899	50.60	1
2013	12	Month Total	999.90	99
2013	Year Total	.....	44444.90	999
2013	1	1152	99.98	2
2013	1	1153	69.00	3
2013	1	1155	564.95	12
. . . rows omitted to save space here				
2013	4	Month Total	222222.21	1000
2013	Year Total	.....	555555.15	3578
Grand Total	.....	.....	999999.90	9999

Task 03: Continuing with the same calculations, display the total lines only for the year totals.

		NumberOrders	AmntDue	NumbBooksPurch
2011	Yearly Total	123	444.00	300
2012	Yearly Total	456	666.00	400
2013	Yearly Total	500	800.00	600
2014	Yearly Total	620	900.00	700
Grand Total	Yearly Total	1699	2910.00	2000

Task 04: This is an author sales report. Rollup total sales (total amount due) for each book by author. Include rows only for authors who have books and the author is the first listed author. Include books that were free. We have some books which have no orders; display the message "No Sales" in the last column if there were no orders for a book and display 0 for the quantity.

Note that there is a grand total line at the bottom with a label; also there are labels for the book total lines. This report is ordered by author id and book id.

AuthorID	BookID	TotalQuantity	TotalSales
F1233	1948	46	123.45
F1233	All books	46	123.45
G6543	142	8	100.25
G6543	143	5	50.25
G6543	144	1	78.25
G6543	All books	14	228.75
H5820	1478	0	No sales
H5820	All books	0	No sales
H7512	2013	25	0.00
H7512	3013	200	0.00
H7512	All Books	225	0.00
. . .			
All Authors	All Books	1205	78978.56

Task 05: Modify the query for task 05. This is still a rollup total sales report.
Now we want to include authors who have no books and books with no authors (use '_anon' as the author id in that case*)* Include the other rows as in task 04

P3002	no books	0	No Sales
P3002	All Books	0	No Sales
. . .			
_anon	5854	9	524.00
_anon	5855	0	No Sales
_anon	All Books	258	10000.06
All Authors	All Books	5252	123123.45

Task 06: Display the books we have, ranked by their page count. We want to use a 50-page range for ranking the page counts. This means that any book with page counts 500-549 have the same rank; page counts 550-599 have the same rank; page counts 600-649 have the same rank; page counts 650-699 have the same rank, etc. In the sample display, we have four books which are all at rank 3. Note that we are not skipping rank numbers- examine the first two sample rows. Do not make assumptions about the maximum page count value. When designing the logic of the query, do not use the current set of data in the table to make decision about the logic.

Book_ID	Page_count	Rank
1001	2000	1
1587	1300	2
8546	982	3
8415	980	3
1474	976	3
1444	970	3
1524	918	4
2005	894	5
200	879	5
201	850	5
4574	825	6
6584	825	6

Task 07: For each day in the month September 2013 display the date, the total quantity of books ordered and the total sales. If there are no sales on a particular date, then display 0 and 0.00 in the second and third columns. The report is displayed in order _date order. You will have 30 rows in the result set.

Sample display layout .

OrderDate	QuantityOrdered	TotalSales
2013-09-01	75	2233.00
2013-09-02	2	84.40
2013-09-03	3	100.25
2013-09-04	0	0.00
2013-09-05	150	1234.20
2013-09-06	0	0.00
2013-09-07	10	567.00