Due Date: April 06, 2106 11:00 pm Points: 35 point max

General Directions - READ THIS

These General Directions apply to all assignments using the AcmeBooks tables.

This assignment uses the tables associated with the AcmeBooks database. There is a pdf file explaining the AcmeBooks database with the sql for the create and inserts. It is important that you read this before you try to do this assignment. That pdf includes a section on general rules and definitions for the tables. For example- it defines the book categories.

Tasks may ask you to filter for book topics in defined sets of values called **categories**; use the definitions of the categories in the pdf file for the books tables. A book topic is not the same thing as a book category

There are a few things about book orders that might not be obvious but they make a difference in writing queries. Suppose we have the following detail rows for an order with order id 98766

```
select *
from bkorders.order_details
where order_id in (98766);
```

order_id	order_line	book_id	quantity	order_price
98766	1	1628	1	32.00
98766	2	1629	1	19.95
98766	3	1162	10	32.00
98766	5	1162	1	19.95

There are 4 detail rows for this order using order_line numbers 1,2,3,5. There is no order_line value of 4 for this order. That is OK. The PK for this table is (order_id, order_line), so we cannot have two rows with the same values for those two columns, but we can skip order_lines.

Notice that book 1162 appears on order_line 3 and on order_line 5 with different values for the order price. That is also ok. (maybe if you buy 10 copies of the book at 32.00, you can get another copy at a lower price.)

The **total number of books** purchased on this order is 13: ; the **total number of different books** purchased on this order is 3 (1628, 1629, 1162). Those are two different questions.

The **total amount due** for this order is $391.90 (**sum(quantity * order_price)**). That is the only correct way to calculate the total amount due for any grouping of books. This formula is used for the total sales (or orders) for a book, the total sales for a date range, the total sales for a customer, the total sales for customers living in California and the total sales for customers in California who ordered book 1162 last year.

For question that refer to the current year, previous year, last year etc, do not use a literal date; derive the value from the system date.

Variables: Some of the tasks may be easier if you use variables. If you use variables, include a **single** Select statement that displays **the values of all variables used in that task**. Variable names must be meaningful. Do not assign data from a table into a variable- we have not discussed that technique. Each task must define all of the variables that it uses within that task sql; redefine the variable if you need to use it for more than one query. Do not define variables at the top of the script to be used later.

The sample display are to show the column order, aliases and formats only. The data values do not reflect the data in the supplied scripts. The values in the sample display may not be consistent between tasks.

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? Think about the difference between count(order_id) and count(distinct order_id) if you are using the order details table.

- Do you need inner join or outer joins? (Hint- **you will not need a full join**; you will lose points if you use 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 columns 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 in the books document, is not going to help your assignment score.

Tasks

Task 01: Find the total amount due for all orders for book id 1108.

```
+--------+
| AmtDue |
+--------+
| 852.30 |
+--------+
```

Task 02: Find the total number of orders for book id 1108 that were placed in the previous year.

```
+-----------------+
| NumberOfOrders |
+-----------------+
|              21 |
+-----------------+
```

Task 03: For each customer we have in the customers table, display the customer id and last name and the number of books they have ordered and the number of books they have ordered in the current year.

```
+--------+--------------+-----------+----------------------+
| CustID | CustName     | BookTotal | CurrentYearBookTotal |
+--------+--------------+-----------+----------------------+
| 200368 | Blake        |      1110 |                  605 |
| 202958 | Denver       |      NULL |                 NULL |
| 208950 | Adams        |        20 |                   10 |
| 211483 | Carroll      |       102 |                 NULL |
| 218709 | Bonnard      |       326 |                  169 |
| 221297 | Dodgson      |        54 |                   13 |
| 222477 | Rossetti     |       145 |                   70 |
| 224038 | Austin       |       111 |                   56 |
```

Task 04: Display the number of orders we had in the previous quarter and the number of customers we have who had at least one order in the previous quarter. The term "previous quarter " means any date in the quarter before the current quarter. This query produces a single row with two columns.
If you run the query in Aug 2014, that is the third quarter of 2014. The query will return data for the second quarter of 2014. If you run the query in Feb 2015, that is the first quarter of 2015, the query will return data for the fourth quarter of 2014.

Task 05: Which book with the category Data Storage Techniques has the most orders? We are using the number of orders for the book - not the quantity. Display the book id and title. Consider there might be ties for first place- in that case all tied books should be returned.

Task 06: Use the cross tab techniques described in the notes for this task.
We want to know how many books we have in the books table in each of the indicated topic **categories**. Display the result as a single output row. The categories to use are:

- Science
- Database Systems
- Data Storage Techniques

There is also a column for all books.
If a book is in more than one category, then it counts in each of those categories.

Science	Database Systems	Data Storage Techniques	All Books
0	456	58	1295

Task 07: For each customer that we have in the customer table, list the customer id and last name and the date of the most recent order for that customer.
If the customer has no orders, then display "No Orders' in the column for the order date.
For this query, if there are order headers that have no associated order detail rows, those orders should not be included.

CustID	CustName	MostRecentOrder
200368	Blake	2016-07-28
212921	Books on Tap	2016-03-22
217002	Grieg	No Orders
217796	Anders	2016-05-06
221297	Dodgson	2016-05-06

Task 08: Display the customer id and last name for customers who have ordered more than $500 worth of books with a topic of SQL but who have never order book id 1142.