

Due Date: Oct 28, 2015 11:00 PM (Late date Oct 29, 11:00 pm)

Points: 35 point max

General Directions

Use the tables for the books database. We are looking for customers with certain patterns of ordering books. The output is limited to orders in the prior year and is based on the month in which books were ordered. The definition of an order is having a row in the order headers table, it does not require any related order detail rows.

Do not use variables in this assignment.

This assignment focuses on the use of the Set operators. So if you do not use the Set operators for tasks 2-8 they will get no points. There are other ways to create these queries- but if you want the points, then follow the directions. **In the sql that you write for the tasks, you are not allowed to use an inner join, outer join, full join or subquery.** The first thing I will do when grading this assignment is a word search for "join"- if you use a join your query gets no points. This also applies to the use of the comma join or a subquery to implement join logic for these tasks. (Your view, described in the following, will contain joins- that is not a problem- you do not write joins in your queries.)

In the first task you are to create a view named **BkOrdersPriorYear** that you will use in the rest of the tasks. You should read through the assignment first and then develop a view that makes the rest of the assignment easier to code.

The output for tasks 2- 8 will all have this format. These should not have any duplicate rows.

```
C_ID  C_NAME
-----
00124 Adams, Samuel
00311 Keats, John
01007 Bonnard, Paul
```

Something to think about regarding the view.

The view needs to expose the customer id. The view needs to expose the customer name as shown in the above sample. You should consider if either or both of the name components can be null and deal with that in the view. The tasks require you to find orders in the previous year- do that in the view. The tasks are concerned with different months, so the view needs to exposed the month so that it can be tested.

Use the fewest table as possible in the view. Deal with the "no duplicate rows" issue in the view.

Do not include anything in the view that is not required- such as a book_id.

Tasks

- Task 01:** Create the view you will use in the rest of the assignment. You will need to use the Oracle syntax for dropping and then creating the view-that is discussed in the notes for this unit.
Use the view name: **BkOrdersPriorYear**
- Task 02:** Display customer id and name for all customers who have at least one order in March and at least one order in June and at least one order in September.
- Task 03:** Display customer id and name for all customers who have at least one order in at least one of the months: April, June, August.
- Task 04:** Display customer id and name for all customers who have at least one order in April and at least one order in May but no orders in June.
- Task 05:** Display customer id and name for all customers who have at least one order in either March or June but have no orders in September.
- Task 06:** Display customer id and name for all customers who have no orders in March or June or September. Since you are using the view as the data source these customers will have some orders in the prior year but not in any of those three months.

- Task 07:** Display customer id and name for all customers who have at least one order in January but no orders in February and no orders in March.
- Task 08:** Display customer id and name for all customers who have at least one order in March or June but not in both March and June.