CECS 323 HOMEWORK: RELATIONAL ALGEBRA 2

OBJECTIVE: How to use the Relational Algebra join operator.

INTRODUCTION: The questions in this assignment all come from the same database that we have been using for practice in the lab. You can find the UML diagram here, and the relation scheme diagram here.

We will use the same Relational Algebra environment that we did in lab. If you need to review how to set that up, you can find the instructions here.

PROCEDURE: Create a Word document that performs the following queries in Relational Algebra.

1. List the names of all Customers that are in the same state as one of our Offices. This will return 26 rows if you do not consider that some countries do not have states, so the state portion of their address will be null.
2. List the names of all Customers who have ordered Products where the vendor is "Classic Metal Creations". This returns 88 rows if you remember to use the "distinct" keyword, which isn't necessary in Relational Algebra of course.
3. List the names of all Customers whose Order was shipped within three days of being ordered. Assume that you can subtract one date from another to get the number of days between two dates in Relational Algebra.

   Sadly, to do this in Derby requires that we resort to a Java escape syntax to perform the necessary date arithmetic. While this looks like voodoo, trust me when I say that it works:

```
select   distinct cust.customerName
from     customers cust inner join orders ord using (customerNumber)
where    abs({fn TIMESTAMPDIFF (SQL_TSI_DAY, ord.ORDERDATE,
             ord.SHIPPEDDATE)}) <= 3;
```

   And it returns 78 rows.

4. List the names of all Customers, their service rep and the Office that the service rep for that customer works in. Returns 122 rows. Take into account that I said that I want to see **all** customers.
5. List the Employee first and last name, and their Customer's name **even if** the Employee is **not** working with a Customer. Returns 108 rows.
6. List all the possible statuses for an order. Returns 6 rows. But this is a little suspect since it really just represents the statuses that we have in use in the data. There might be other statuses possible, but without a table of valid statuses to govern what goes into the status column, it's impossible to say.
7. List the Order Number for all Orders in the 'On Hold' status where the quantity of a product ordered is greater than the quantity of that product on hand. This returns 0 rows. Apparently, the 'On Hold' requirement removes any rows that we might have gotten otherwise.

8. List the Employee LastName and FirstName that work in Japan. This returns 2 rows.
9. List the productLine, the ProductName and the quantityOrdered for all products ordered during the month of July. You will need a sigma that checks for month(orderDate) = ‘July’. This gets you 164.
10. List the customerName, the paymentDate, and the amount on all payments that exceeded $1000. This returns 272 rows.
11. List the productLine for all products ordered by customers from the State of ‘Louisiana’. Sadly, there are no customers from this fine state. Plus, the state in the Customers table is the two-character state code, not the state name. Sorry about that.

WHAT TO TURN IN:

- For each problem give me:
  - The original question.
  - The Relational Algebra statement.
  - A screenshot of the graphical representation of the Relational Algebra.
- Put all your output into a single Word document and upload that into the dropbox.
- Your team’s filled out collaboration document. You can find a template for that here.