**POINTS: 10**

* Unless otherwise specified, write any query needing a join in the ANSI format.
* Do not assume you can use variables or parameterized queries (if you’re already familiar with them.)
* Make sure each query has a comment above to indicate the problem number and the query should be formatted as follows:

--9

SELECT col1

,col 2

,col 3

FROM table1

WHERE condition1

AND/OR condition2

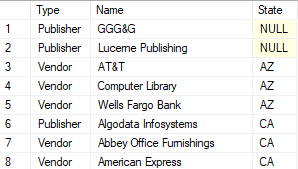
Murach Chapter 4 with the following modifications :

1. Set the context to the AP database.
2. Murcach Problem 1 – Modify to show only vendors from New York
3. Murcach Problem 2 – Modify it to sort it from highest to lowest balance.
4. Murcach Problem 8 – Modify it to extend it to include New York as a state of interest. For those not in CA or NY, use “Neither” for the state. Results would be similar to:

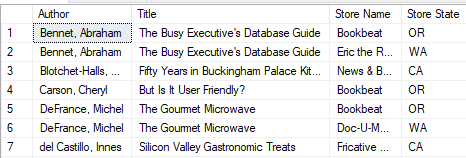


Additional problems using the pubs database also:

1. Switch context to the pubs database.
2. Create a list of all the authors who have a phone number with the area code 801 (show all columns).
3. Produce a list that shows the first name, last name and titles of their books. Make sure authors who have never published show up as well. (Note: you will need to look at using title\_author)
4. It just so happens that the book company uses the same vendors in the AP database. Create a list that shows publishers from the pubs database and vendors from the AP database. The list should show what type of entity (i.e. Publisher or Vendor), the name of the company and the state the company is in. The results will look similar to the following:



1. Examine the links that connect authors to the stores where their titles are sold. Write a query that shows the sale of every author’s title and the store sold in. The query should show the title, author’s name in the format of “Lastname, FirstName” and the name of the store and the state the store is located in. Further, since some titles are sold more than once, make sure we see each unique row only once. Here is an example of the output:



1. Change context to the AdventureWorks2017 database. Take time to study the relationships among the tables in the HumanResources Schema.
2. Human resources wants to do a review of each employee’s pay history. Create a list that shows each employee ID, the employee login ID, the rate they were paid with each pay rate change and the date of the pay rate change. (You will need the HumanResources.Employee and HumanResources.EmployeePayHistory tables) It should be sorted with the highest rates first, and with only the four highest rates showing.
3. One more request. HR wants a list of those employees who might not have an Address yet. This also needs to include the Employee’s login ID. (You need to determine the appropriate tables to use based on your knowledge primary and foreign keys.)