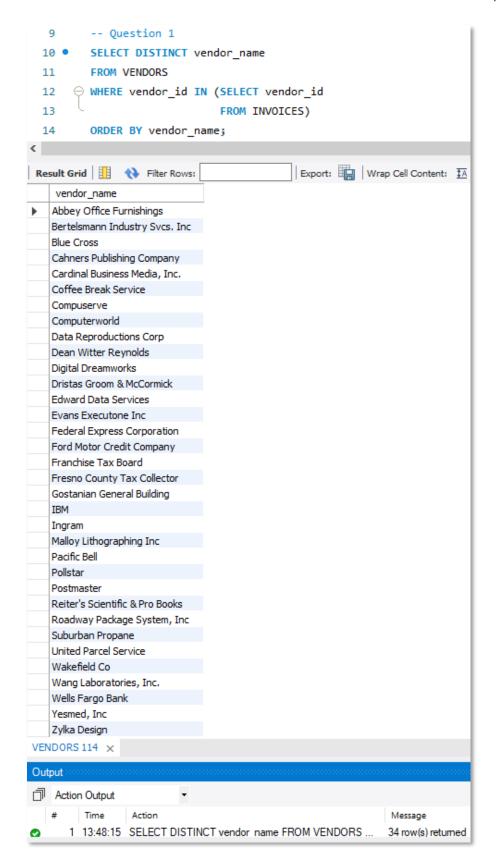
## **HOMEWORK 4**

(10 points) Write a SELECT statement that returns the same result set as this SELECT statement, but don't use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

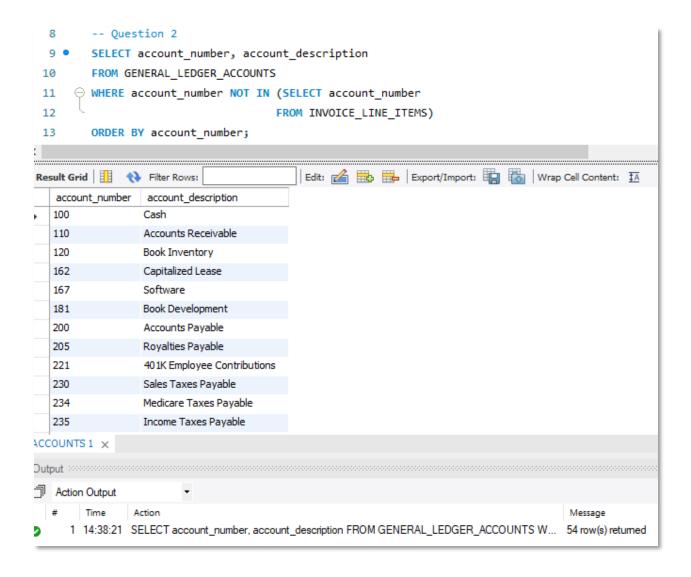
(Answer on next page)



 (10 points) Write a SELECT statement that returns two columns from the General Ledger Accounts table: account number and account description.

Return one row for each account number that has never been assigned to any line item in the Invoice\_Line\_Items table. To do that, use a subquery introduced with the NOT EXISTS operator.

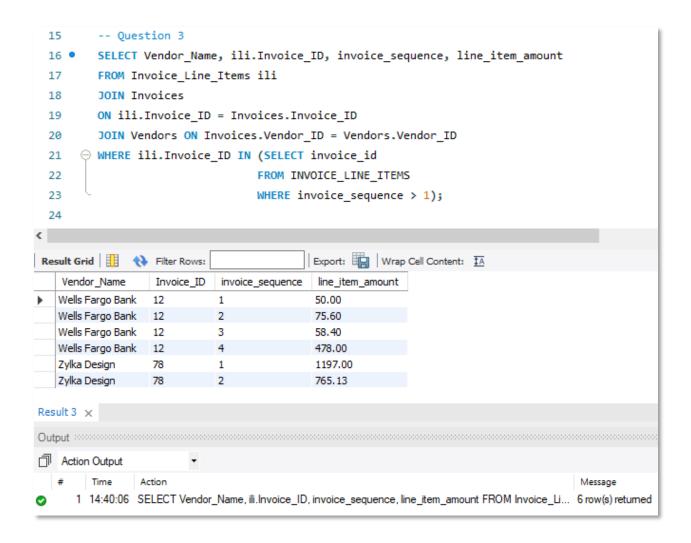
This should return 54 rows. Sort the results by the account number column.



(10 points) Write a SELECT statement that returns four columns: vendor\_name, invoice\_id, invoice sequence, and line item amount.

Return a row for each line item of each invoice that has more than one line item in the Invoice\_Line\_Items table. Hint: Use a subquery that tests for invoice\_sequence > 1. This should return 6 rows.

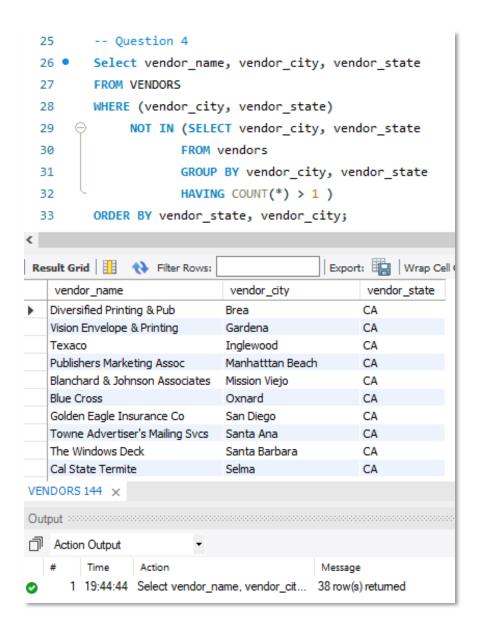
Hint: Before embarking on writing the query, it would help if you first navigated the database tables in question and first try to answer the question through navigation.



4. (10 points) Write a SELECT statement that returns the name, city, and state of each vendor that's located in a unique city and state. In other words, don't include vendors that have a city and state in common with another vendor. This should return 38 rows.

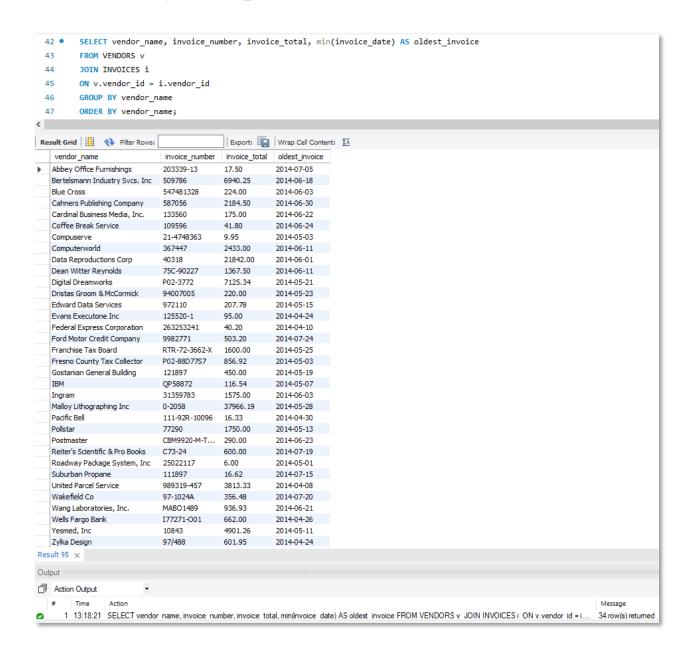
Sort the results by the vendor state and vendor city columns.

Hint: For this question, first create a subquery containing a list of vendor state and vendor city that are repeated (COUNT > 1). Then, write the main query whose results are NOT IN the results from the subquery.

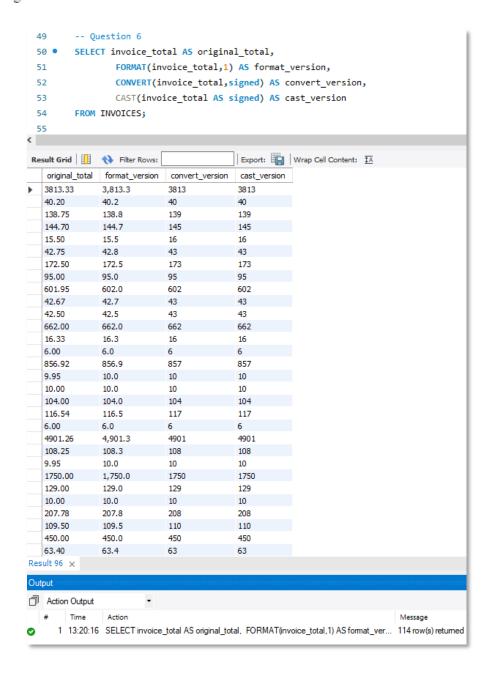


5. (10 points) Write a query to return one row per vendor, representing the vendor's oldest invoice (the one with the earliest date). Each row should include these four columns: vendor name, invoice number, invoice date, and invoice total. This should return 34 rows.

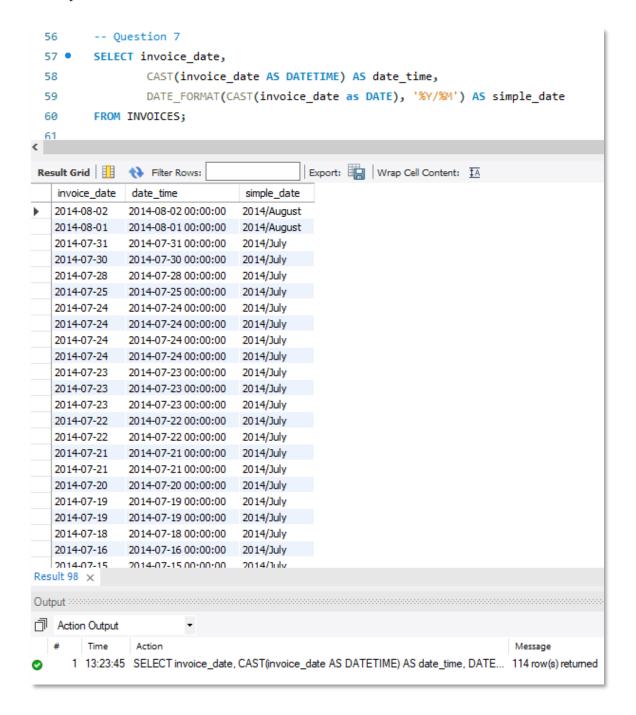
Sort the results by the vendor name column.



- 6. (10 points) Write a SELECT statement that returns these columns from the Invoices table:
  - The invoice total column.
  - A column that uses the FORMAT function to return the invoice\_total column with 1 digit to the right of the decimal point.
  - A column that uses the CONVERT function to return the invoice\_total column as an
    integer. A column that uses the CAST function to return the invoice\_total column as an
    integer.

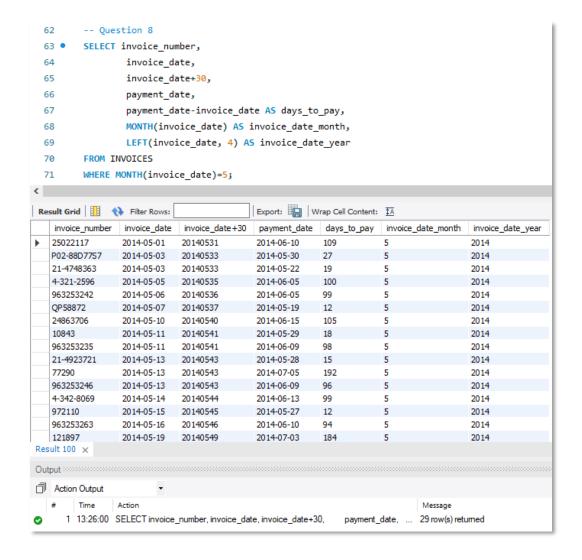


- 7. (10 points) Write a SELECT statement that returns these columns from the Invoices table:
  - The invoice date column.
  - A column that uses the CAST function to return the invoice\_date column with its full date and time.
  - A column that uses the CAST function to return the invoice\_date column with just the
    year and the month.



- 8. (10 points) Write a SELECT statement that returns these columns from the Invoices table:
  - The invoice number column
  - The invoice date column
  - The invoice date column plus 30 days
  - The payment date column
  - A column named days\_to\_pay that shows the number of days between the invoice date and the payment date
  - The number of the invoice date's month
  - The four-digit year of the invoice date

When you have this working, add a WHERE clause that retrieves just the invoices for the month of May based on the invoice date, not the number of the invoice month. The query should return 29 rows.

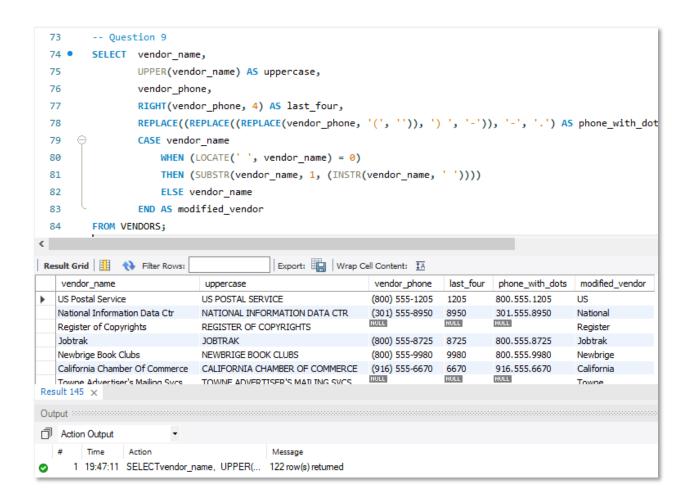


Page 9 of 16

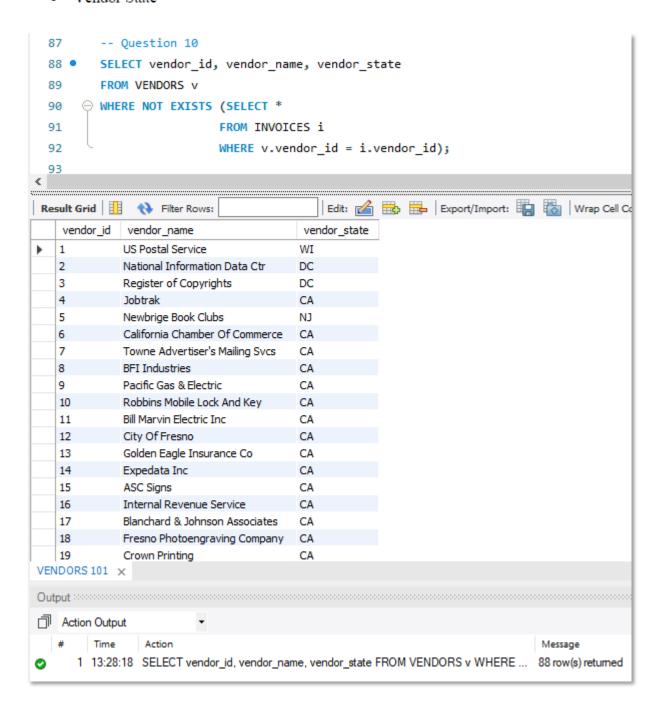
- 9. (10 points) Write a SELECT statement that returns these columns from the Vendors table:
  - The vendor name column.
  - The vendor name column in all capital letters.
  - The vendor phone column.
  - A column that displays the last four digits of each phone number.

When you get that working right, add the columns that follow to the result set.

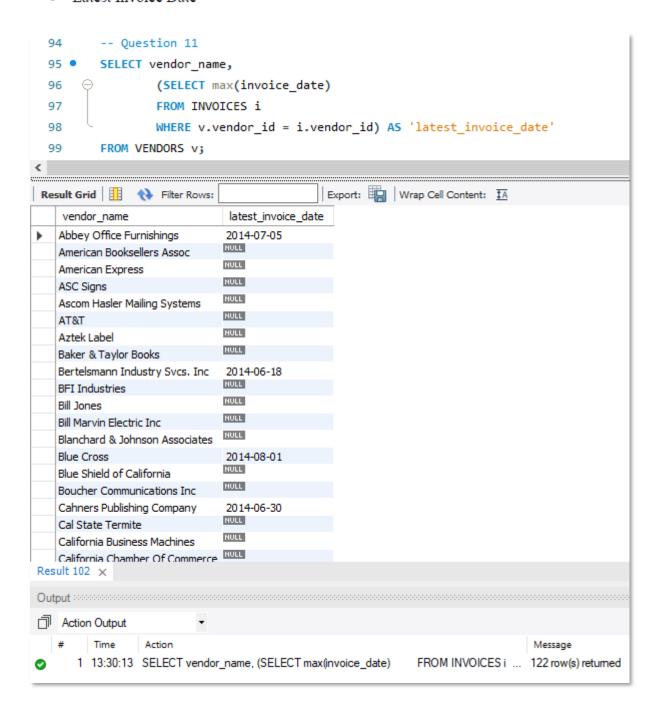
- The vendor\_phone column with the parts of the number separated by dots, as in 555.555.5555. Hint: For this, you can use the REPLACE function in MySQL.
- A column that displays the first word in each vendor name if there is a space in the vendor\_name and display the full vendor\_name if there isn't a space in the vendor\_name.



- 10. (10 points) Display all vendors that don't have invoices. Use NOT EXISTS. The result set should contain 88 rows. The result set should contain the following columns:
  - Vendor ID
  - Vendor Name
  - Vendor State

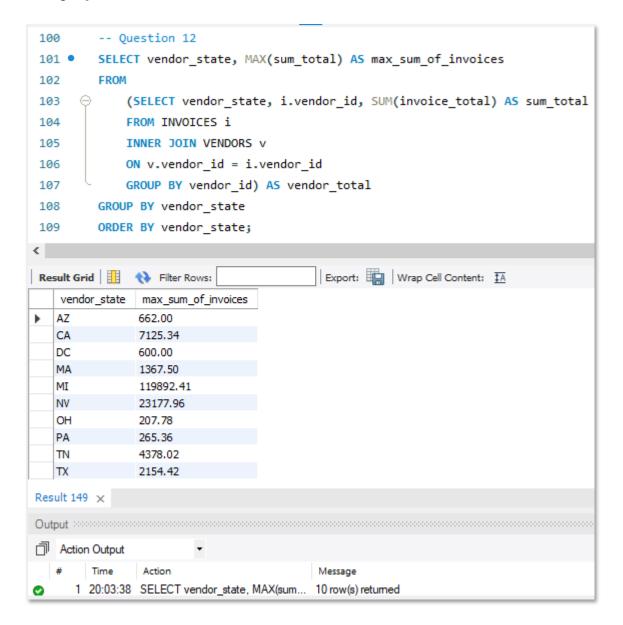


- 11. (10 points) Get the most recent invoice date for each vendor. Use subqueries and NOT joins. The query should return 122 rows. The result set should contain the following columns:
  - Vendor Name
  - Latest Invoice Date



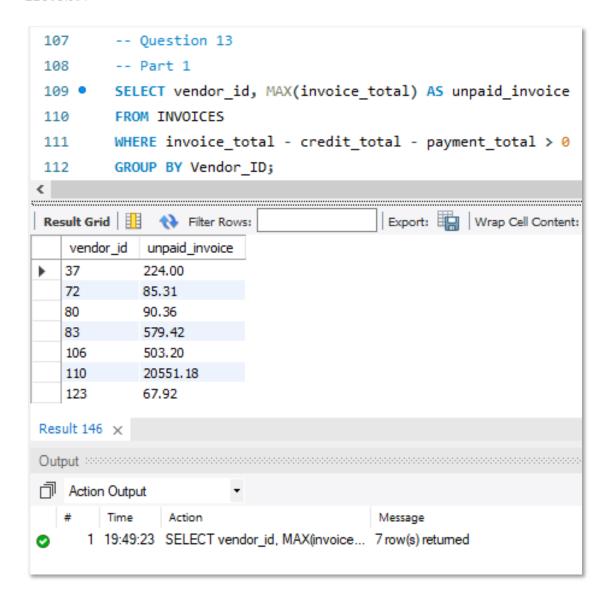
- 12. (10 points) Get the largest invoice total for the top vendor in each state. Largest invoice total means MAX(sum of invoices). Display the following two columns in your resultset:
  - Vendor State
  - Max Sum of Invoices

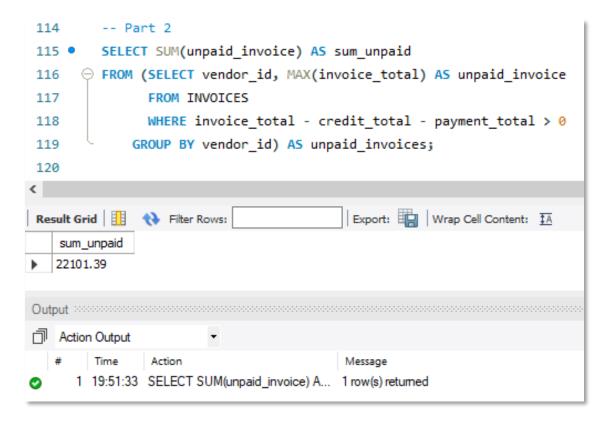
The query should return 10 rows.



13. (20 points) Write a SELECT statement that returns two columns: vendor\_id and the largest unpaid invoice for each vendor. To get this, you can group the result set by the vendor\_id column. This should return 7 rows.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return a single value that represents the sum of the largest unpaid invoices for each vendor. If your query is correct, then you should get this single value as 22101.39.





14. (10 points) Write a SELECT statement that answers this question: Which invoices have a payment total that's greater than the average payment total for all invoices with a payment total greater than 0?

Return the invoice\_number and invoice\_total columns for each invoice. This should return 20 rows.

Sort the results by the invoice total column in descending order.

```
121
                                       -- Question 14
                                        SELECT invoice number, invoice total
  122 •
  123
                                        FROM INVOICES

→ WHERE payment total > (SELECT AVG(payment total)
  124
  125
                                                                                                                                                      FROM INVOICES
  126
                                                                                                                                                      WHERE payment total > 0)
                                        ORDER BY invoice total DESC;
  127
                                                                                                                                                                                         Export: Wrap Cell
  Result Grid
                                                                 Filter Rows:
               invoice_number
                                                                           invoice_total
             0-2058
                                                                         37966, 19
             P-0259
                                                                         26881.40
             0-2060
                                                                         23517.58
             40318
                                                                         21842.00
             P02-3772
                                                                         7125.34
             509786
                                                                         6940.25
             10843
                                                                         4901.26
                                                                         3813.33
             989319-457
             989319-447
                                                                         3689.99
             989319-437
                                                                         2765.36
             367447
                                                                         2433.00
             989319-467
                                                                         2318.03
INVOICES 148 ×
Output reconcesses conservations are conservations and the conservation of the conserv
Action Output
                                    Time
                                                                      Action
                                                                                                                                                                                       Message
                         1 19:55:22 SELECT invoice_number, invoic... 20 row(s) returned
```