**Use ap database to complete Exercises 1-8 at the end of Chapter 7 in your book.**

**\*Screen clip your query and results below. DO NOT screen print all when the result sets has many rows … just the first few and last few rows including the row count. (12.5 points each)**

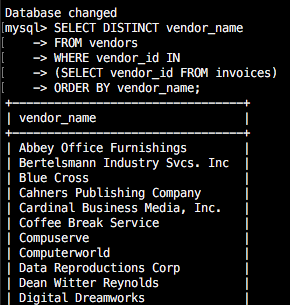
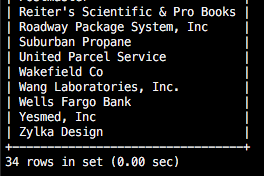
1a. Write a SELECT statement that returns the same result set as this SELECT statement, but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

SELECT DISTINCT vendor\_name

FROM vendors JOIN invoices

ON vendors.vendor\_id = invoices.vendor\_id

ORDER BY vendor\_name;

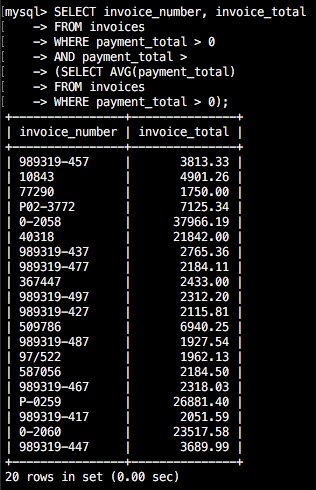
 

1b. For the SELECT statement in Q1, what is the goal of that statement?

To list all vendors with invoices

2. 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?

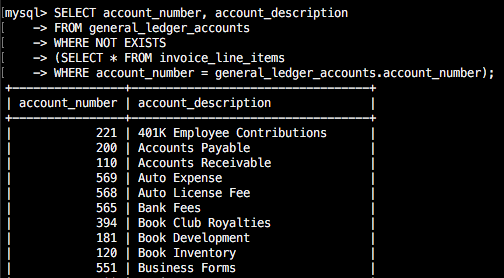
Return the invoice\_number and invoice\_total columns for each invoice. This should return 20 rows.

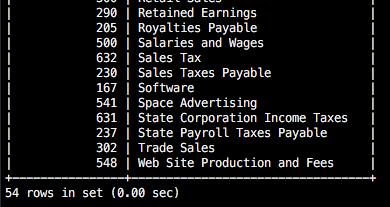


3. Write a select statement that returns two columns from the general\_ledger\_accounts table: account\_number and account\_description.

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.

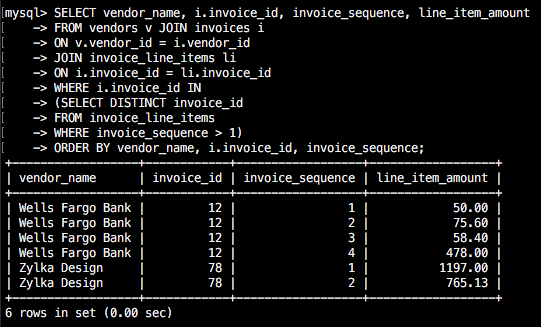
Sort the results by the account\_number column.





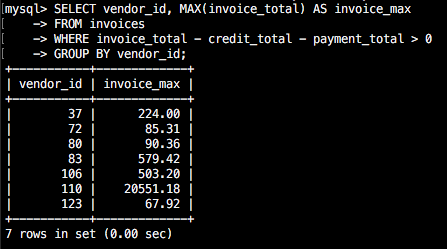
4. Write a SELECT statement that returns four columns: vendor\_name, invoice\_id, invoice\_sequence, and line\_item\_amount.

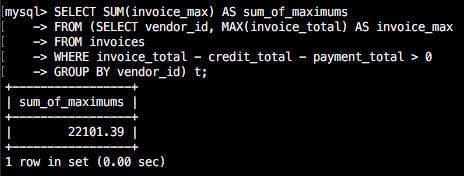
Return a row for each line item of each invoice that has more tha one line item in the invoice\_line\_items table. *Hint: Us a subquery that tests for invoice\_sequence > 1.* This should return 6 rows.



5. Write a SELECT statement that returns two columns: vendor\_id and the largest unpaid invoice for each vendor. To do 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 it’s FROM clause. The main query should return a single value that represents the sum of the largest unpaid invoices for each vendor. This should return 1 row.





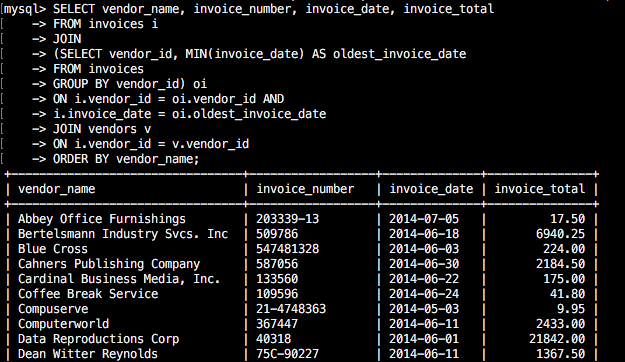
6. Write a SELECT statement that returns the name, city, and state of each vendor that’s located in a unique city and state. In other words, don’t include vendors that have a city and state in common with another vendor. This should return 38 rows.

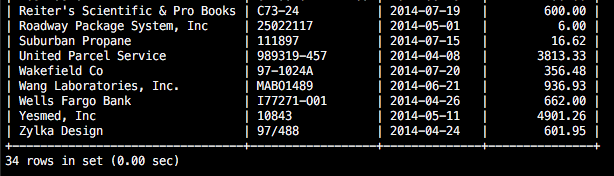
Sort the results by the vendor\_state and vendor\_sity columns.

7. Use a correlated subquery 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.

8. Rewrite Q7 so it gets the same result but uses an inline view instead of a correlated subquery.



****