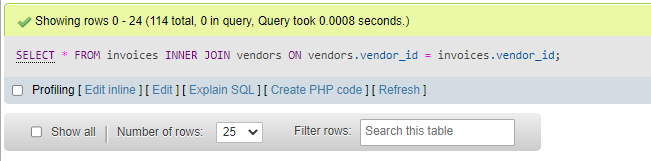
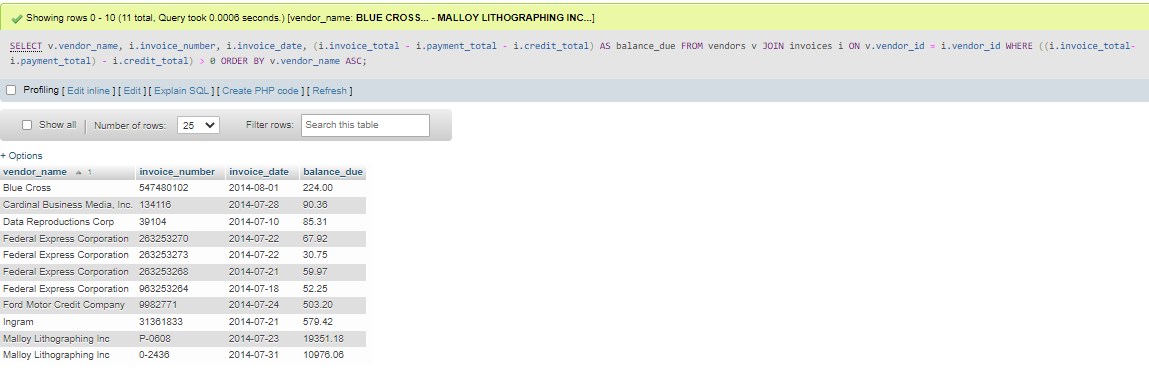
Mp02SQL\_Styles

**PART I**

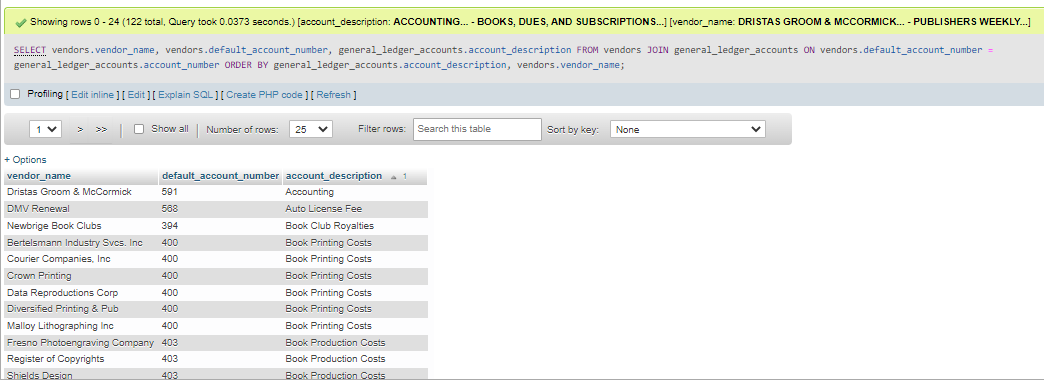
1. Selects all tuples from the invoices table that has matching vendor\_ids in the vendors table



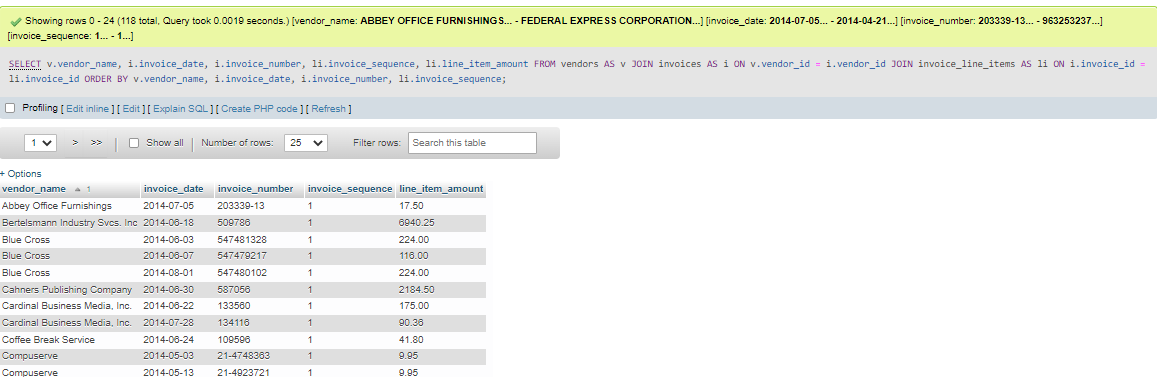
1. Selects vendor\_name, invoice\_number, and balance\_due (calculated attribute using data from invoices table) attributes from vendors and invoices tables that have joined tuples based on vendor\_id and only shows the balance\_due amount that’s greater than 0 and is sorted in ascending order by the vendor\_name attribute.



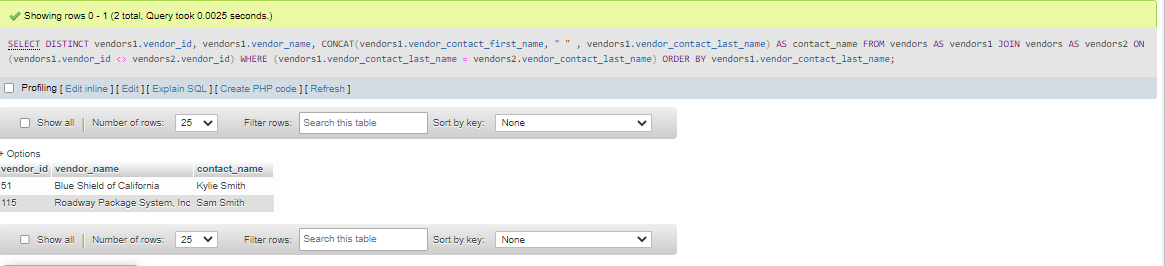
1. Selects vendor\_name, default\_account\_number, account\_description attributes from the vendors and general\_ledger\_accounts table that have joined based on the vendor’s default\_account\_number and general\_ledger\_accounts’ account\_number attributes; finally it is ordered by account\_description, and then by vendor\_name.



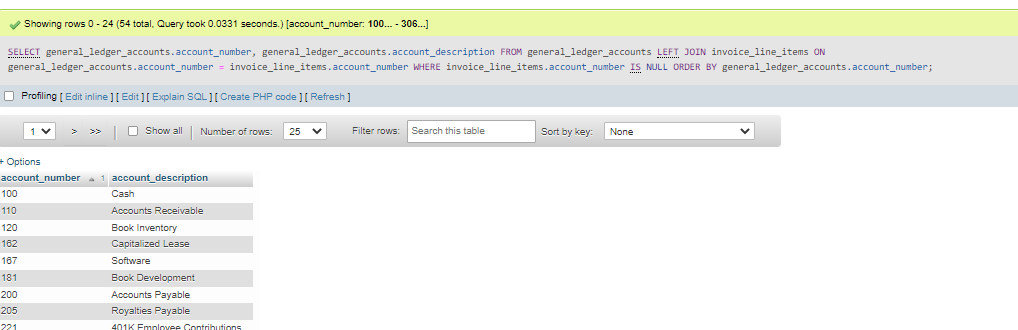
1. Selects vendor\_name, invoice\_date, invoice\_number, invoice\_sequence, line\_item\_amount attributes from the vendors table (with the given alias v), the invoices table (with the given alias i), and the invoice\_line\_items table (with the given alias li). The vendors and invoices tables are joined together based on vendor\_id, afterwards the newly joined table is joined by invoices\_line\_items based on invoice\_id. Finally, the result table is ordered by the vendor\_name attribute, then the invoice\_date attribute, followed by the invoice\_number attribute, and then the invoice\_sequence attribute.



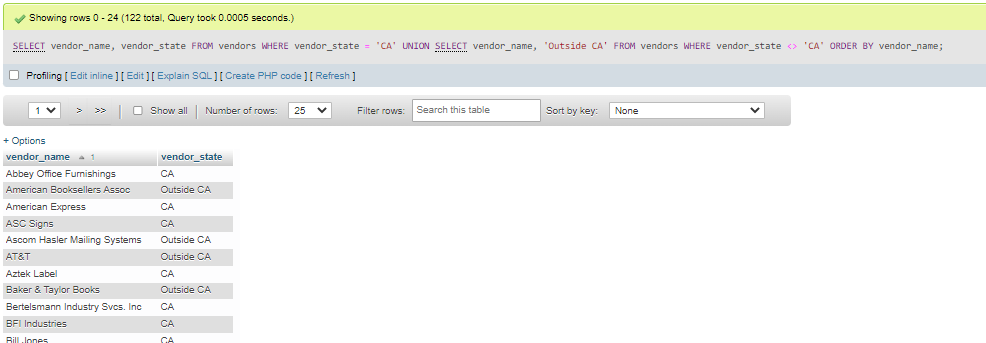
1. Selects only the unique tuples with the attributes vendor\_id, vendor\_name, and a created attribute with the alias contact\_name using concatenated vendor\_contact\_first\_name and vendor\_contact\_last\_name; all of which comes from the vendors table. The vendors table is joined with itself (both instances of vendors table is given an alias to help better differentiate it) based on the vendors table1 vendor\_id attribute not being equal to vendors table2 vendor\_id attribute and where vendors table1 last\_name attribute is equal to vendors table2 last\_name attribute. Finally, the result table is ordered by vendor\_contact\_last\_name attribute.



1. Selects account\_number and account\_description attributes from general\_ledger\_accounts table that joins with invoice\_line\_items table (returns all tuples from general\_ledger\_accounts and any matching records from invoice\_line\_items table) based on account\_number attribute. Afterward it only returns tuples where the invoice\_line\_items account\_number is null. Finally, the result table is ordered by account\_number attribute.

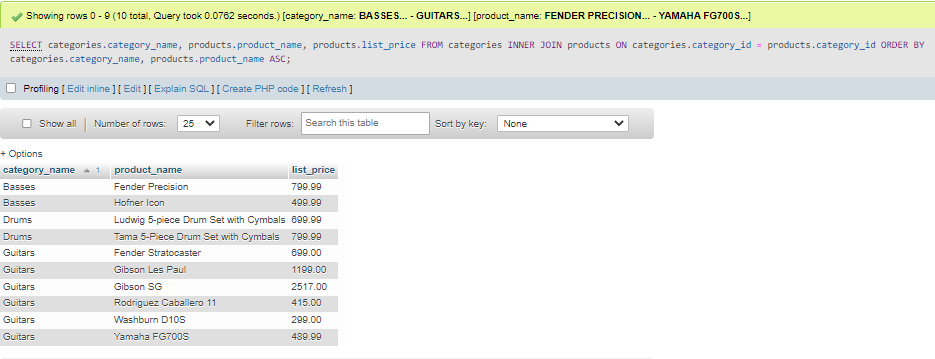


1. Selects vendor\_name and vendor\_state attributes from vendors table and returns only tuples where vendor\_state = “CA”. Skipping ahead a little bit, we also select vendor\_name and vendor\_state attribute which sets all tuples to “Outside CA”, unless the vendor is not equal to “CA”. We then use a union to combine both selection statements and order the result table by vendor\_name.

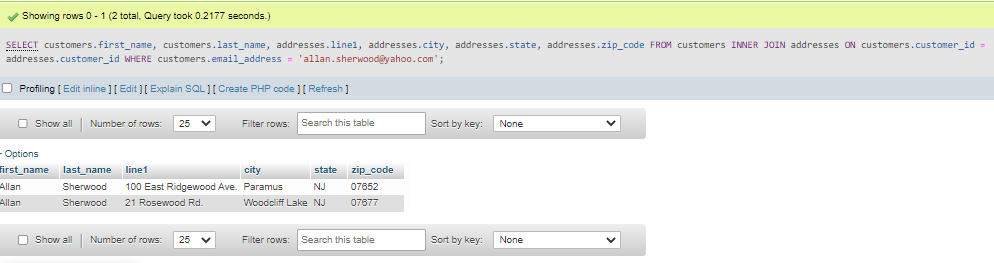


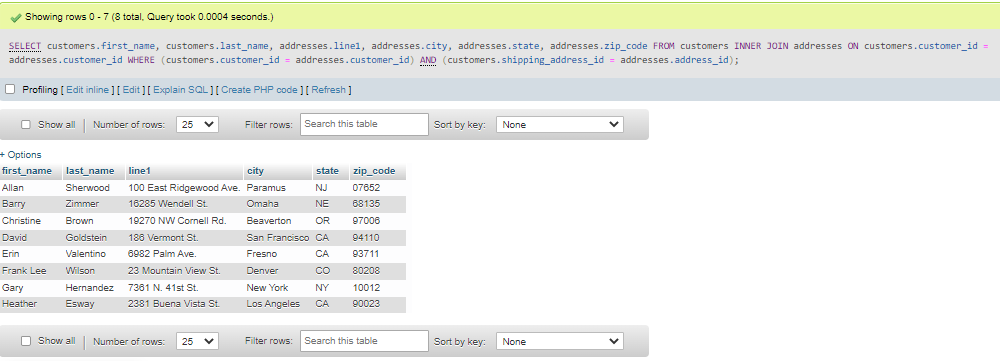
**PART II**

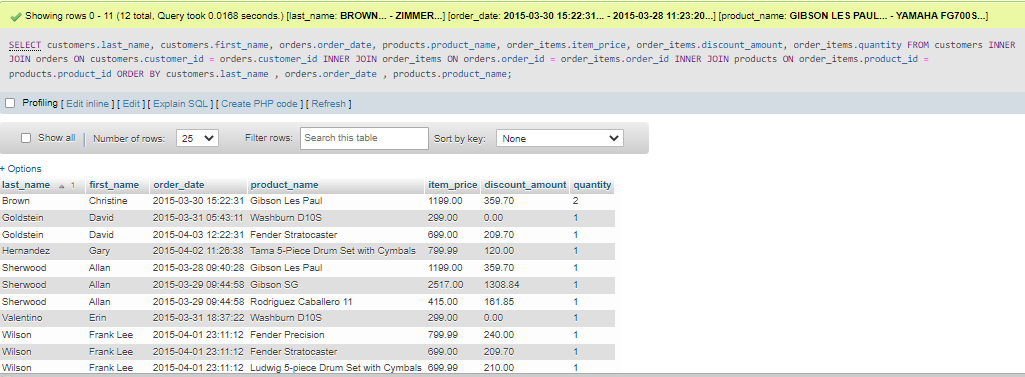
1. Selects category\_name, product\_name, list\_price attributes from categories table which has been joined with the products table based on *matching* category\_id attributes, which is then sorted in ascending order by category\_name attribute, then by product\_name attribute.



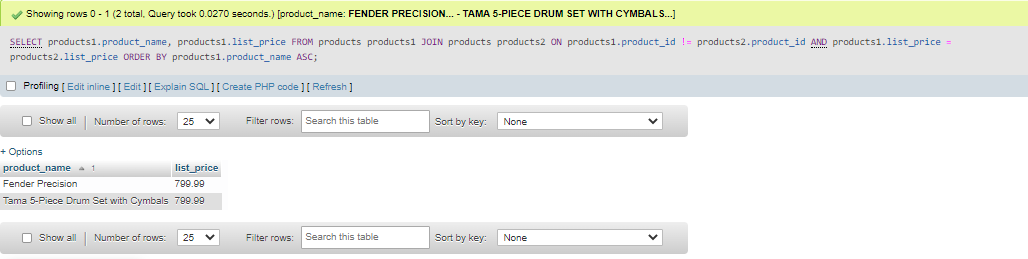
1. Selects first\_name, last\_name, line1, city, state, and zip\_code attributes from the customers table that has joined with addresses table based on *matching* customer\_id attributes from both the addresses and customer tables. Only returns the tuples where the email\_address attribute equals “allan.sherwood@yahoo.com”



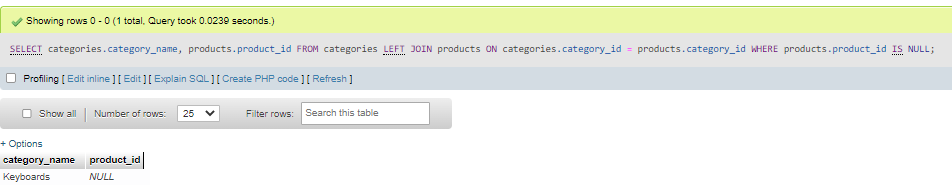
1. Selects first\_name, last\_name, line1, city, state, and zip\_code attributes from the customers table that has joined with addresses table based on *matching* customer\_id attributes from both the addresses and customer tables. Only returns tuples where customers and addresses customer\_id attributes are equal AND the customers shipping\_address\_id attribute and addresses address\_id attribute are equal.
2. Selects last\_name, first\_name, order\_date, product\_name, item\_price, discount\_amount, and quanity attributes from the customer table and orders table which is joined on *matching* customer\_id attributes from both the orders and customer tables; all of which is joined on *matching* order\_id attributes with the order\_items table; which in turn gets joined again on *matching* product\_id attributes with the products table. Finally the result table gets ordered by last\_name attribute, followed by the order\_date attribute, and lastly the product\_name attribute.



1. Select product\_name and list\_price from products which is joined with itself (first products table is assigned the alias products1 with the second products table being assigned the alias products2 to help differentiate them) based on product1 product\_id attribute not equal to product2 product\_id attribute AND product2 list\_price attribute being equal to product1 list\_price attribute



1. Selects category\_name and product\_id attributes from the categories and products tables, both of which have joined returning all tuples from the categories table and any matching tuples from the products table based on both tables category\_id attribute. Only returns tuples where the product\_id attribute from the products table is equal to NULL.



1. Selects the string “Shipped” and the order\_id attribute as newly created attributes for the results table as well as the order\_date attribute, all of which from the orders table where the ship\_date attribute IS NOT NULL. Skipping ahead, another selection is made with the string “Not shipped” and the order\_id attribute for a newly created attribute for the second result table as well as the order\_date from the orders table where the ship date is NULL. Both selection statements are combined so any null statements within the created ship\_status attribute is changed to “Not Shipped" and any statements that are not null are then changed to “Shipped”. Finally the result table is ordered by the order\_date attribute.

