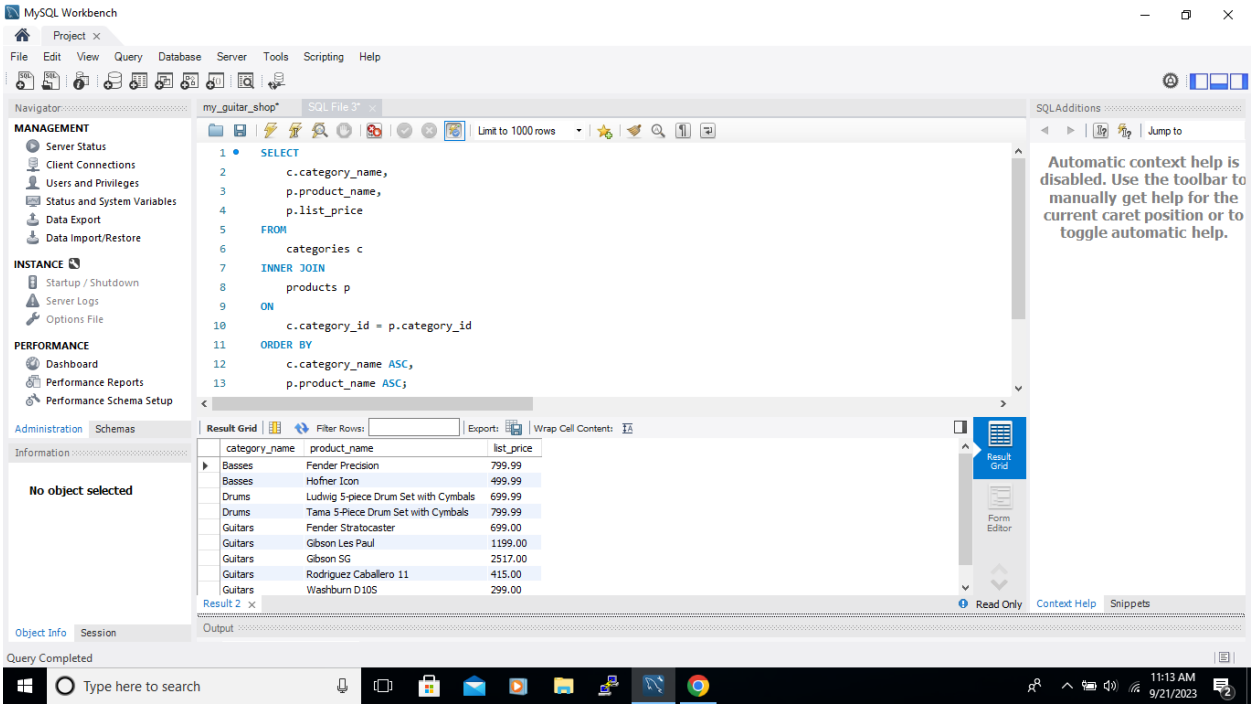
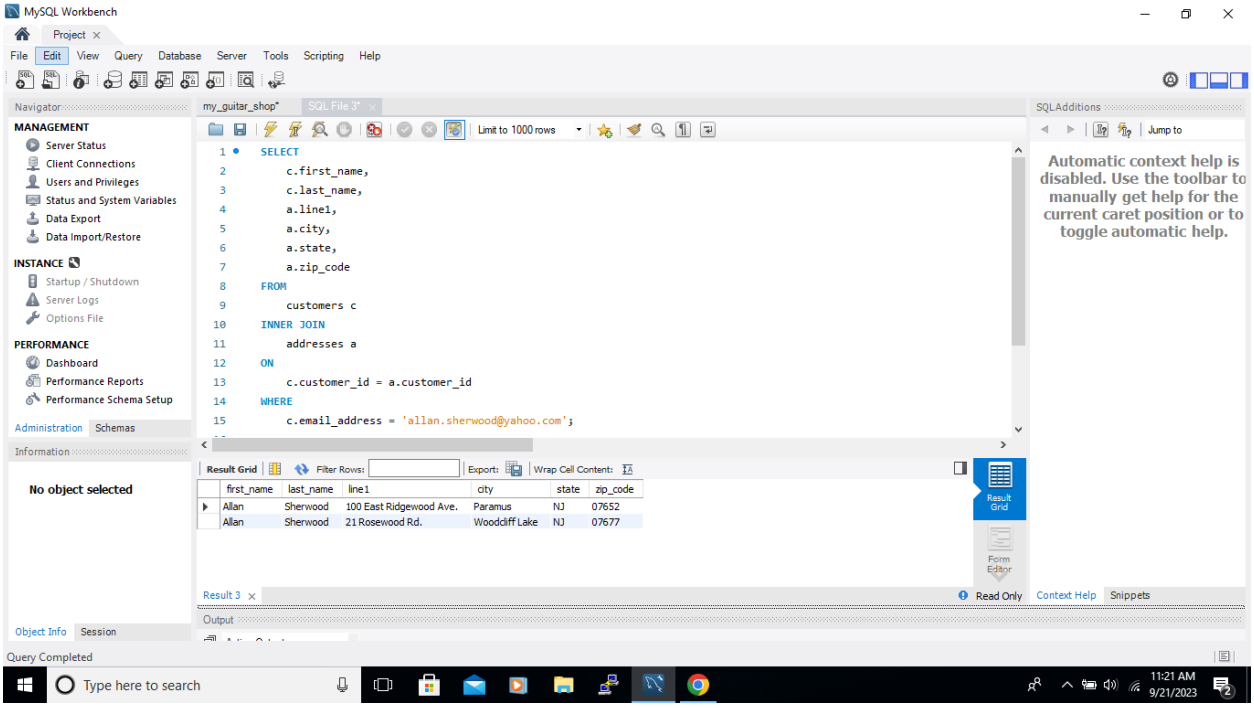
Lab 4: How to Retrieve Data From Two or more Tables

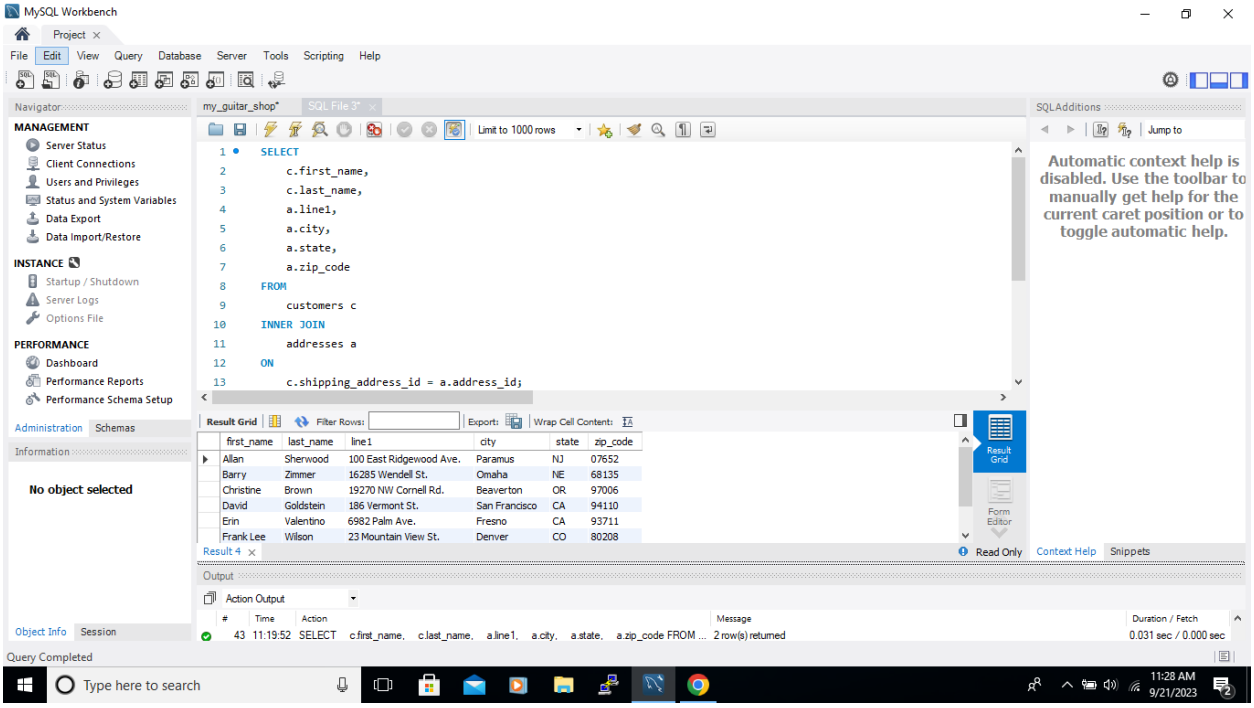
Step 1: Write a SELECT statement that joins the Categories table to the Products table and returns these columns: category_name, product_name, list_price.
Sort the result set by the category_name column and then by the product_name column in ascending sequence.



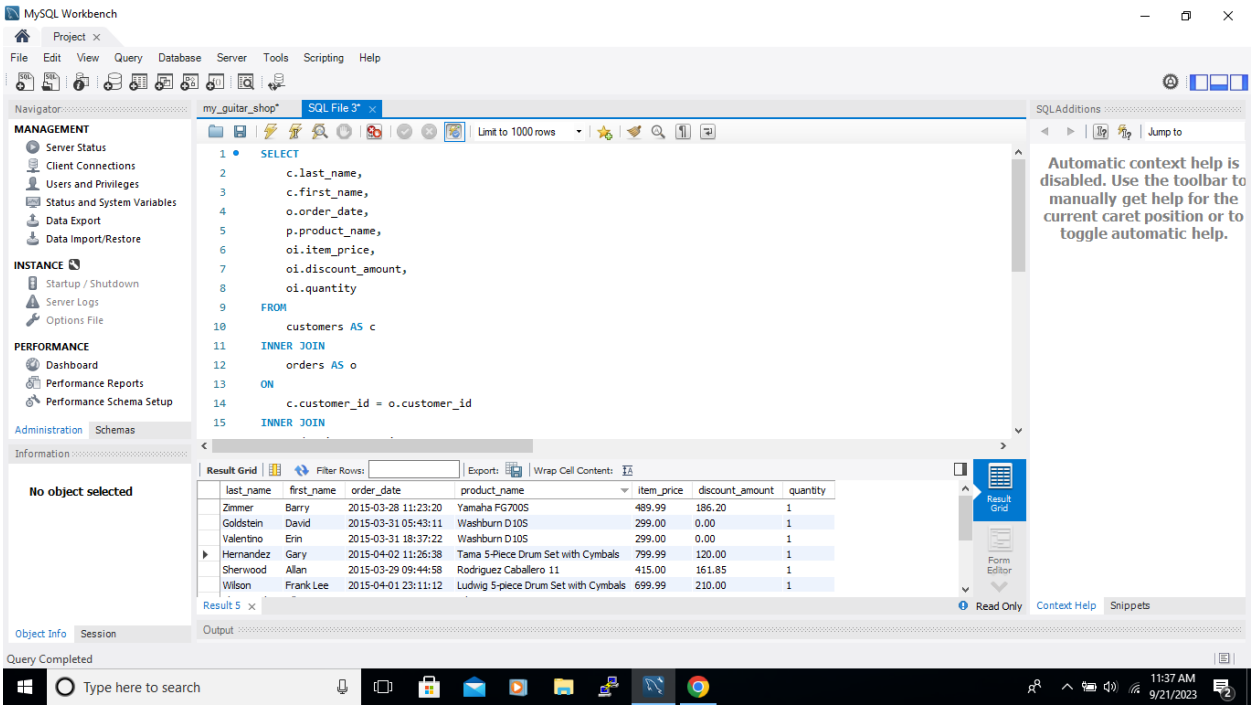
Step 2: Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code. Return one row for each address for the customer with an email address of allan.sherwood@yahoo.com.



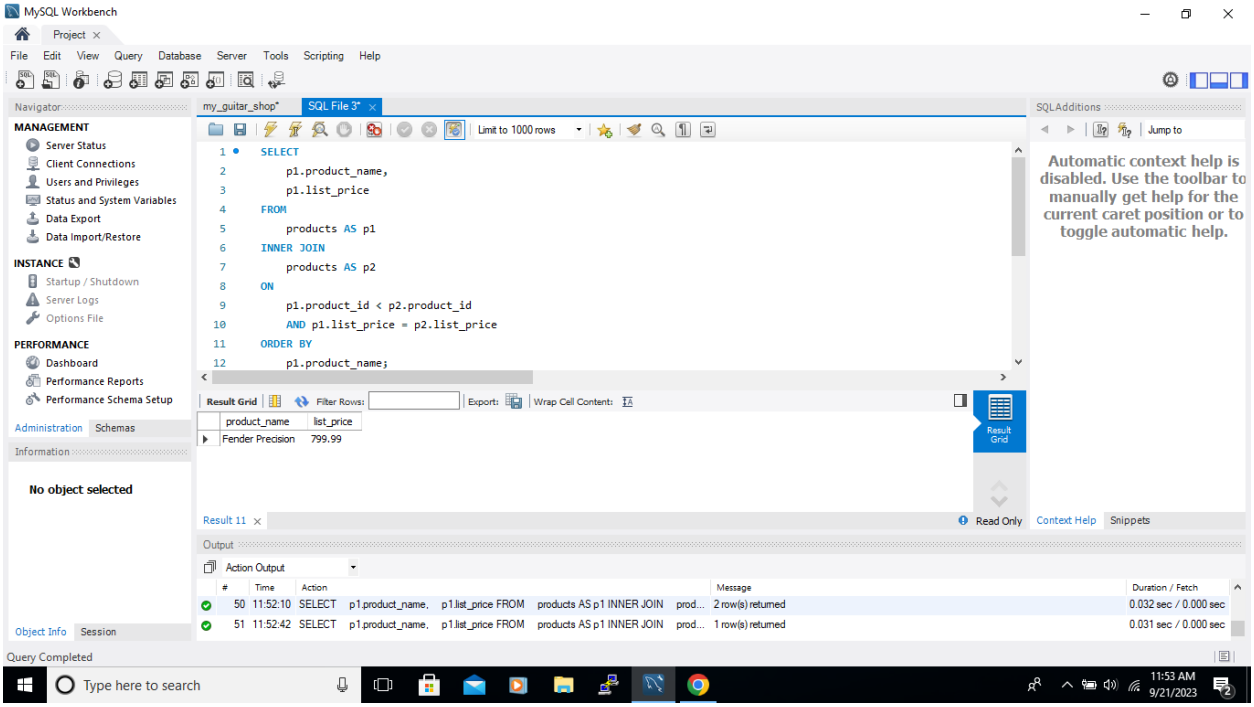
Step 3: Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code. Return one row for each customer, but only return addresses that are the shipping address for a customer.



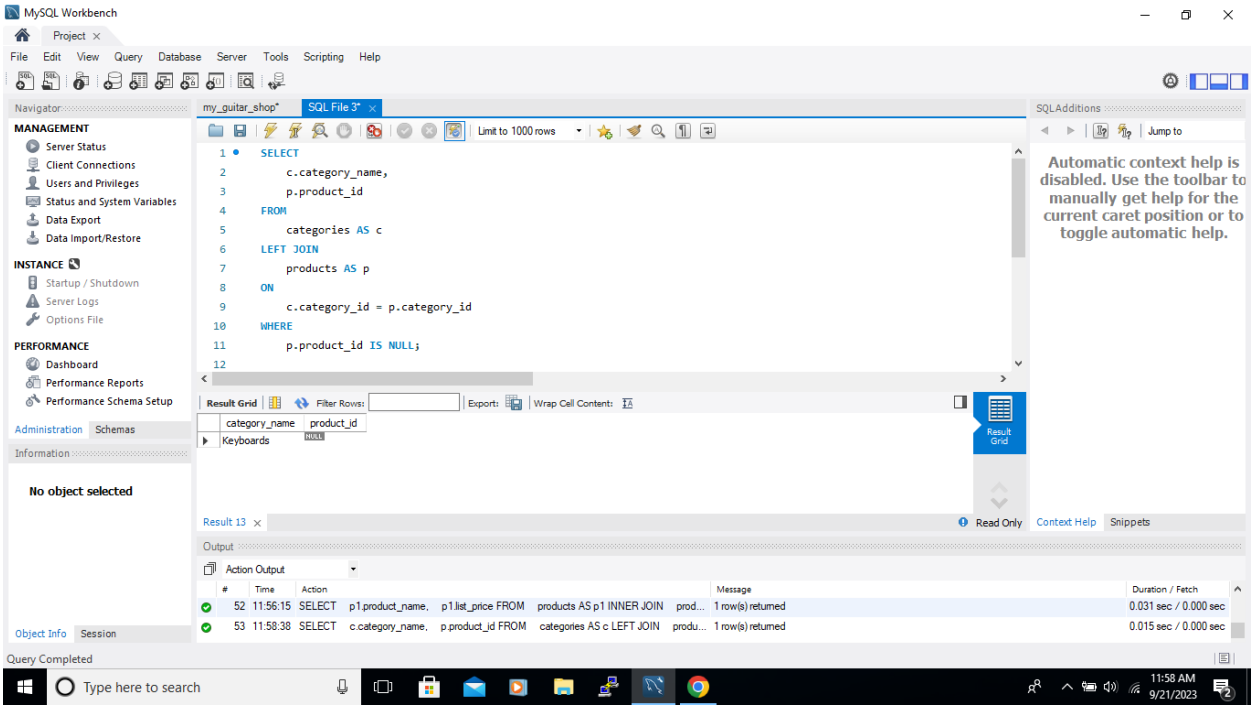
Step 4: Write a SELECT statement that joins the Customers, Orders, Order_Items, and Products tables. This statement should return these columns: last_name, first_name, order_date, product_name, item_price, discount_amount, and quantity. Use aliases for the tables. Sort the final result set by the last_name, order_date, and product_name columns.



Step 5: Write a SELECT statement that returns the product_name and list_price columns from the Products table. Return one row for each product that has the same list price as another product. Hint: Use a self-join to check that the product_id columns aren't equal but the list_price columns are equal. Sort the result set by the product_name column.



Step 6: Write a SELECT statement that returns these two columns: category_name. The category_name column from the Categories table product_id The product_id column from the Products table Return one row for each category that has never been used. Hint: Use an outer join and only return rows where the product_id column contains a null value.



Step 7: Use the UNION operator to generate a result set consisting of three columns from the Orders table: ship_status A calculated column that contains a value of SHIPPED or NOT SHIPPED order_id The order_id column order_date. The order_date column If the order has a value in the ship_date column, the ship_status column should contain a value of SHIPPED. Otherwise, it should contain a value of NOT SHIPPED. Sort the final result set by the order_date column.