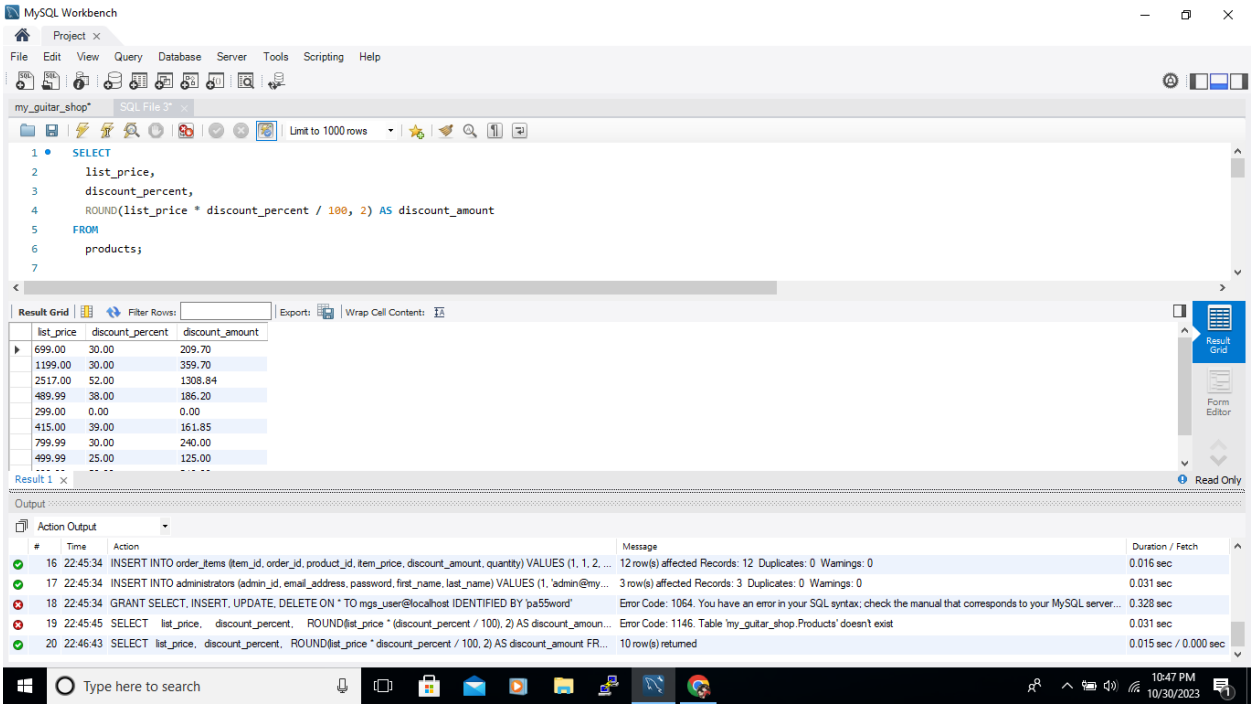
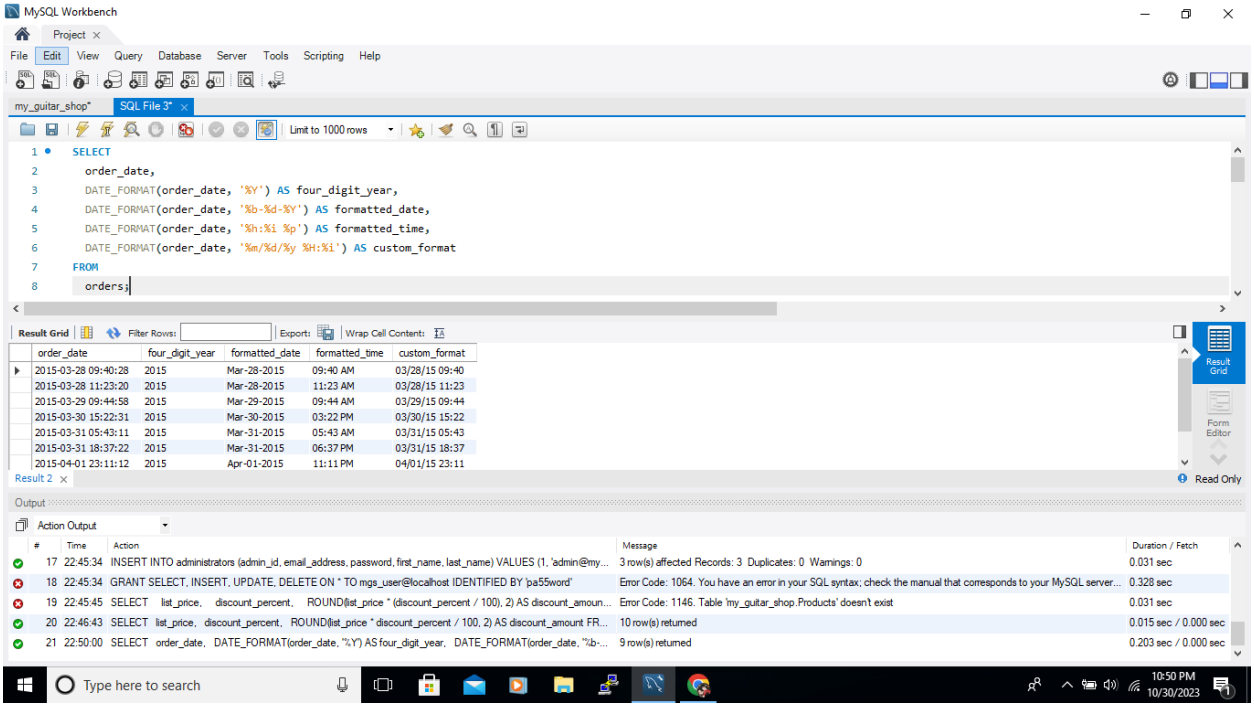
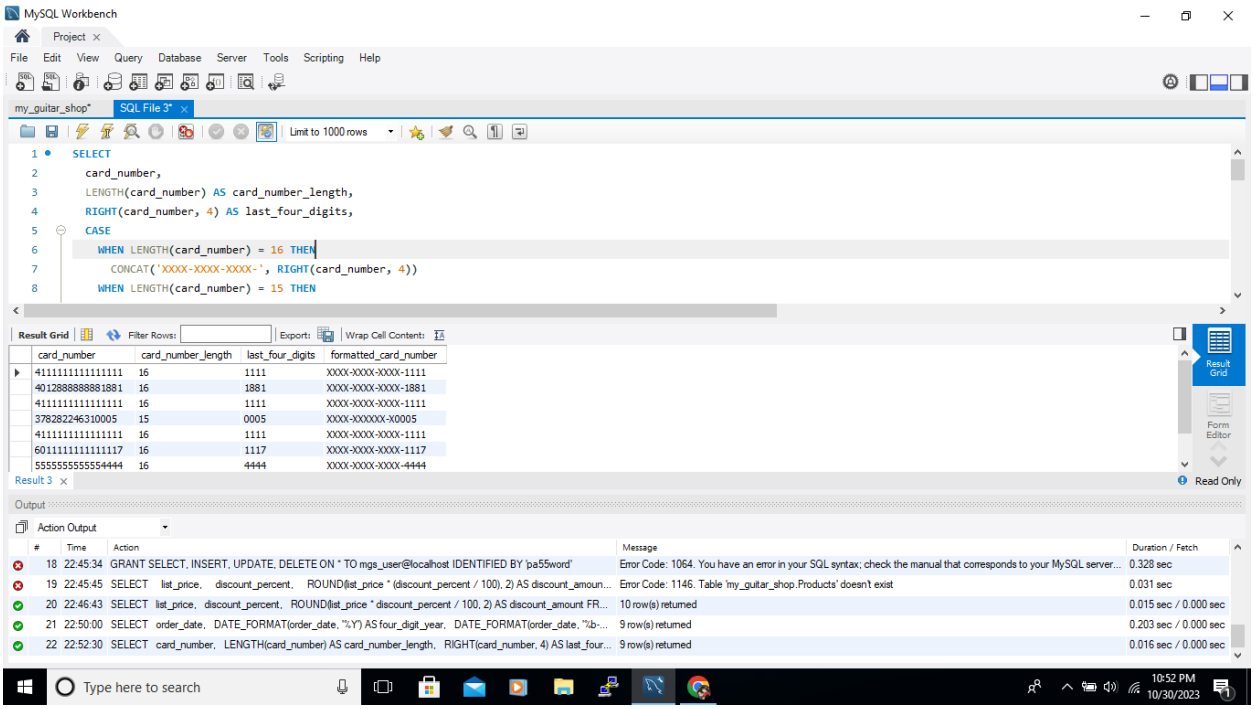
1. Write a SELECT statement that returns these columns from the Products table: The list_price column The discount_percent column A column named discount_amount that uses the previous two columns to calculate the discount amount and uses the ROUND function to round the result so it has 2 decimal digits.



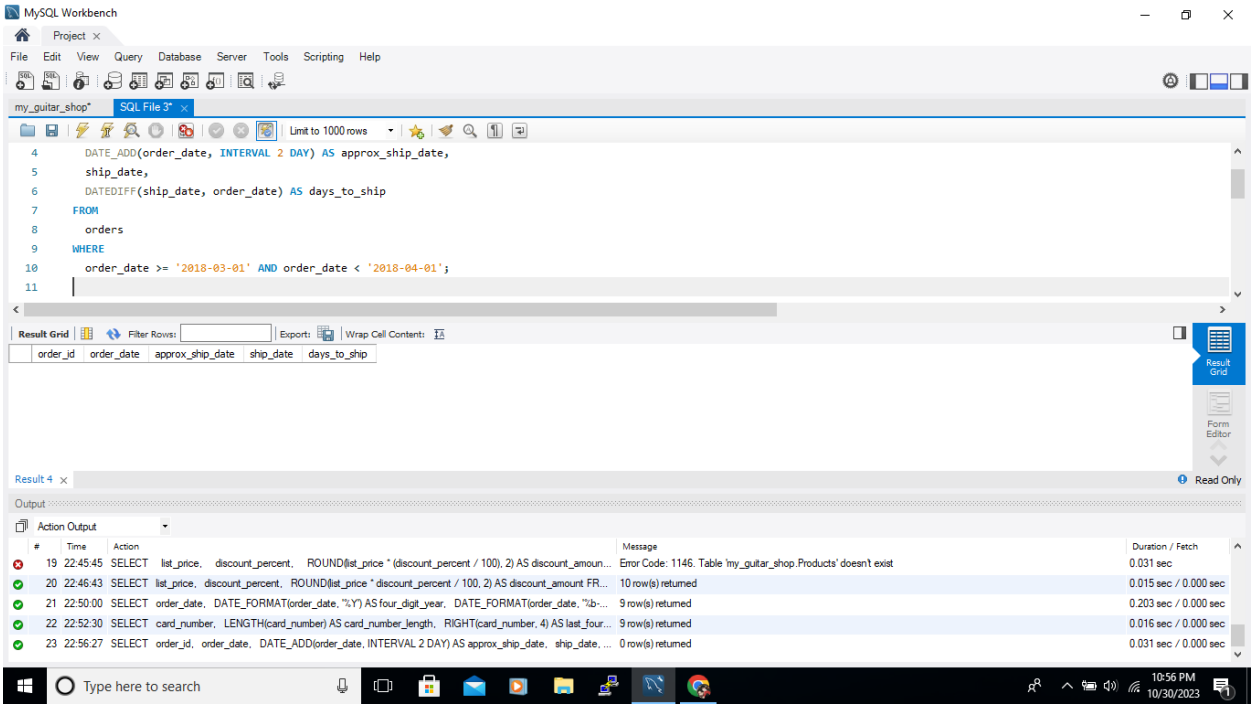
2. Write a SELECT statement that returns these columns from the Orders table: The order_date column A column that uses the DATE_FORMAT function to return the four- digit year that's stored in the order_date column A column that uses the DATE_FORMAT function to return the order_date column in this format: Mon-DD-YYYY. In other words, use abbreviated months and separate each date component with dashes. A column that uses the DATE_FORMAT function to return the order_date column with only the hours and minutes on a 12-hour clock with an am/pm indicator A column that uses the DATE_FORMAT function to return the order_date column in this format: MM/DD/YY HH:SS. In other words, use two-digit months, days, and years and separate them by slashes. Use 2-digit hours and minutes on a 24-hour clock. And use leading zeros for all date/time components.



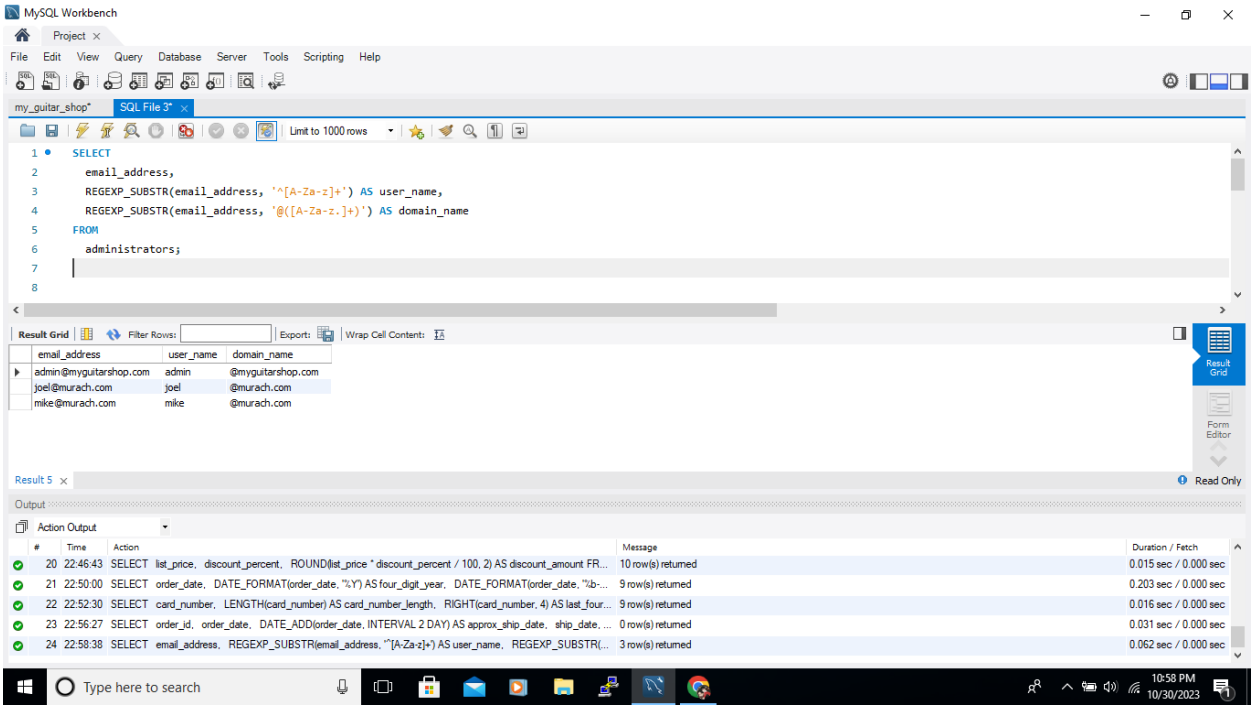
3. Write a SELECT statement that returns these columns from the Orders table: The card_number column The length of the card_number column When you get that working right, add the columns that follow to the result set. This is more difficult because these columns require the use of functions within functions. The last four digits of the card_number column A column that displays an X for each digit of the card_number column except for the last four digits. If the card number contains 16 digits, it should be displayed in this format: XXXX-XXXX-XXXX-1234, where 1234 are the actual last four digits of the number. If the card number contains 15 digits, it should be displayed in this format: XXXX-XXXXXX-X1234. (Hint: Use an IF function to determine which format to use.)



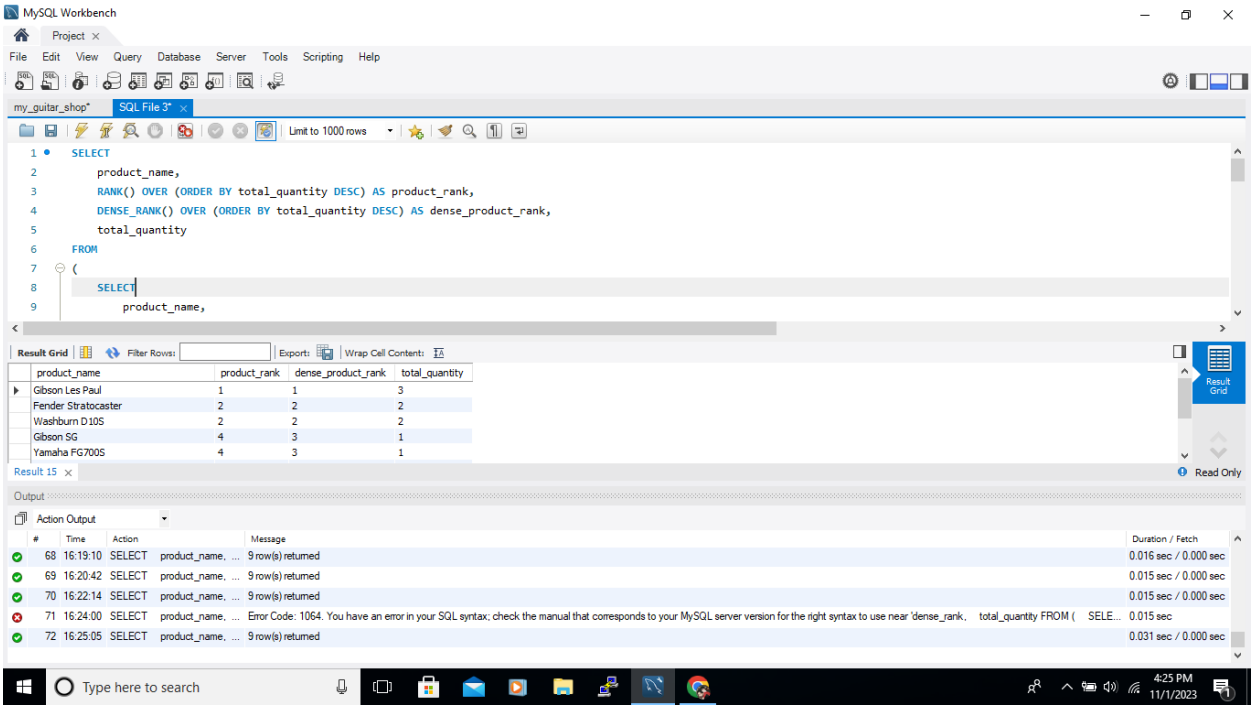
4. Write a SELECT statement that returns these columns from the Orders table: The order_id column The order_date column A column named approx_ship_date that's calculated by adding 2 days to the order_date column The ship_date column if it doesn't contain a null value A column named days_to_ship that shows the number of days between the order date and the ship date When you have this working, add a WHERE clause that retrieves just the orders for March 2018.



5. Write a SELECT statement that uses regular expression functions to get the username and domain name parts of the email addresses in the Administrators table. Return these columns: The email_address column A column named user_name that contains the username part of the email_address column (the part before the @ symbol) A column named domain_name that contains the domain name part of the email_address column (the part after the @ symbol) Note: The username part of the email addresses contains only letters, and the domain name part contains only letters and a period.



6. Write a SELECT statement that uses the ranking functions to rank products by the total quantity sold. Return these columns: The product_name column from the Products table A column named total_quantity that shows the sum of the quantity for each product in the order_items table A column named rank that uses the RANK function to rank the total quantity in descending sequence A column named dense_rank that uses the DENSE_RANK function to rank the total quantity in descending sequence.



7. Write a SELECT statement that uses the analytic functions to get the highest and lowest sales by product within each category. Return these columns: The category_name column from the Categories table The product_name column from the Products table A column named total_sales that shows the sum of the sales for each product with sales in the order_items table A column named highest_sales that uses the FIRST_VALUE function to show the name of the product with the highest sales within each category A column named lowest_sales that uses the LAST_VALUE function to show the name of the product with the lowest sales within each category