Project Portfolio

Jacob Huckleberry

ITS410: Database Management

Colorado State University Global

Dr. Raquel Hicks

August 7, 2022

Figure 1

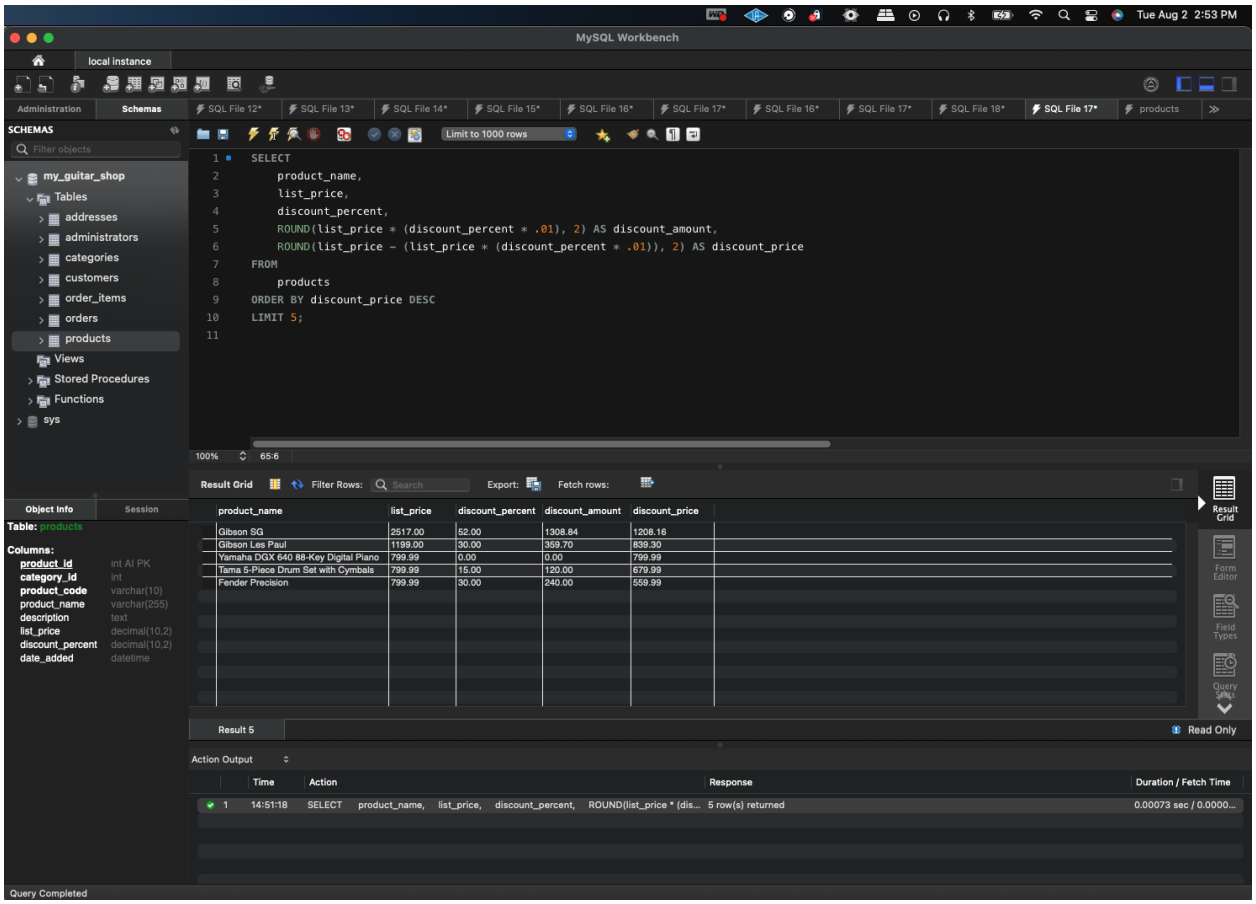

Figure 1 shows a query which selects the product name, list price and discount percent columns from the products table. The query also creates a column called discount_amount which is calculated from the discount percent and list price columns. The query creates another column called discount_price which takes the same calculation from the discount_amount column but subtracts it with the list_price value. Both of these created columns are rounded to the nearest hundredth. The query is in descending order by the discount price column and is limited to return the first five rows.

Figure 2

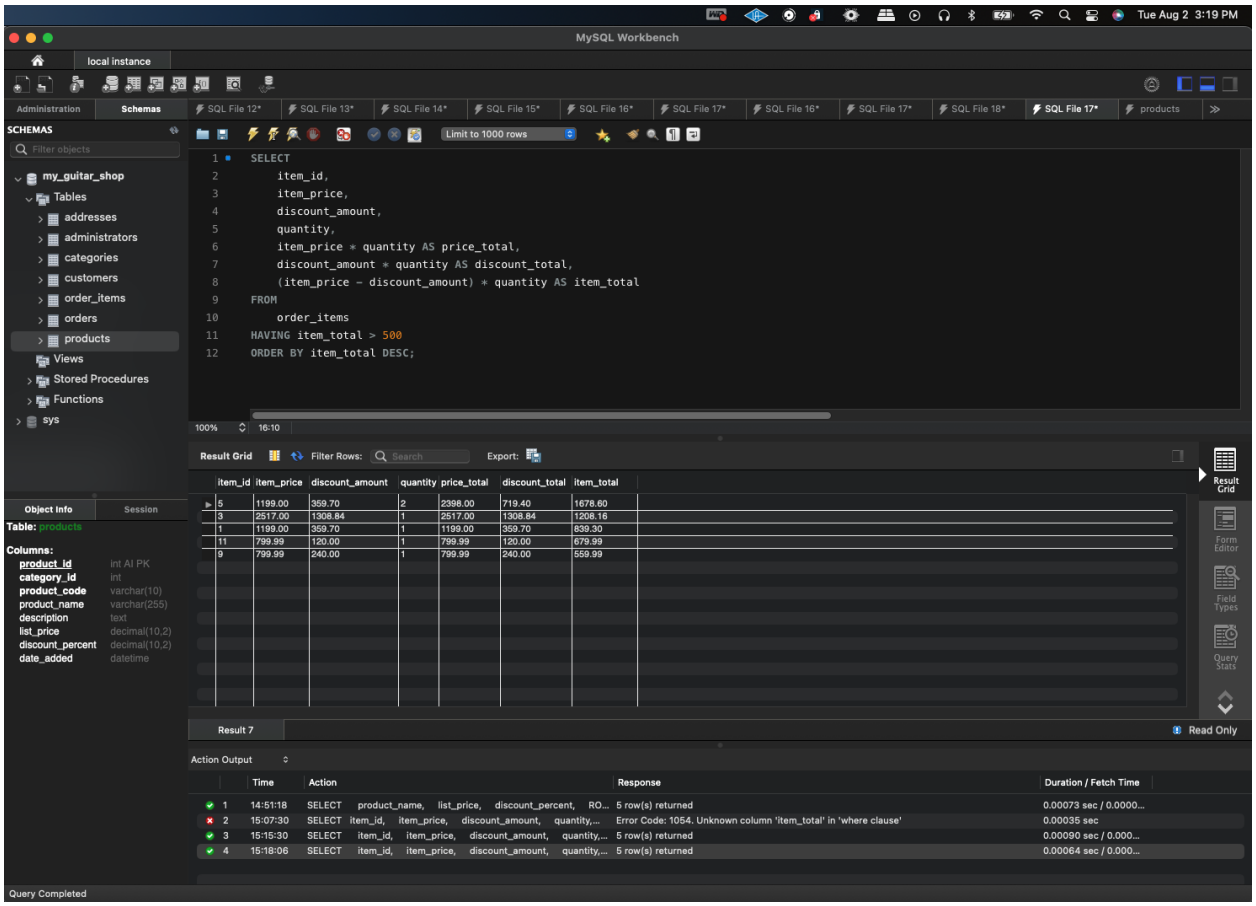

Figure 2 shows a query which selects the item id, item price, discount amount and quantity columns from the order_items table. The query also creates three columns called price_total, discount_total and item_total. The price_total is calculated by the product of the item price and quantity. The discount total is calculated from the product of theitem price and quantity. The item total is calculated from the subtraction of the item price and discount amount all multiplied by the quantity. Lastly, the query is in descending order by the item total and only displays results which are above the item total of 500.

Figure 3

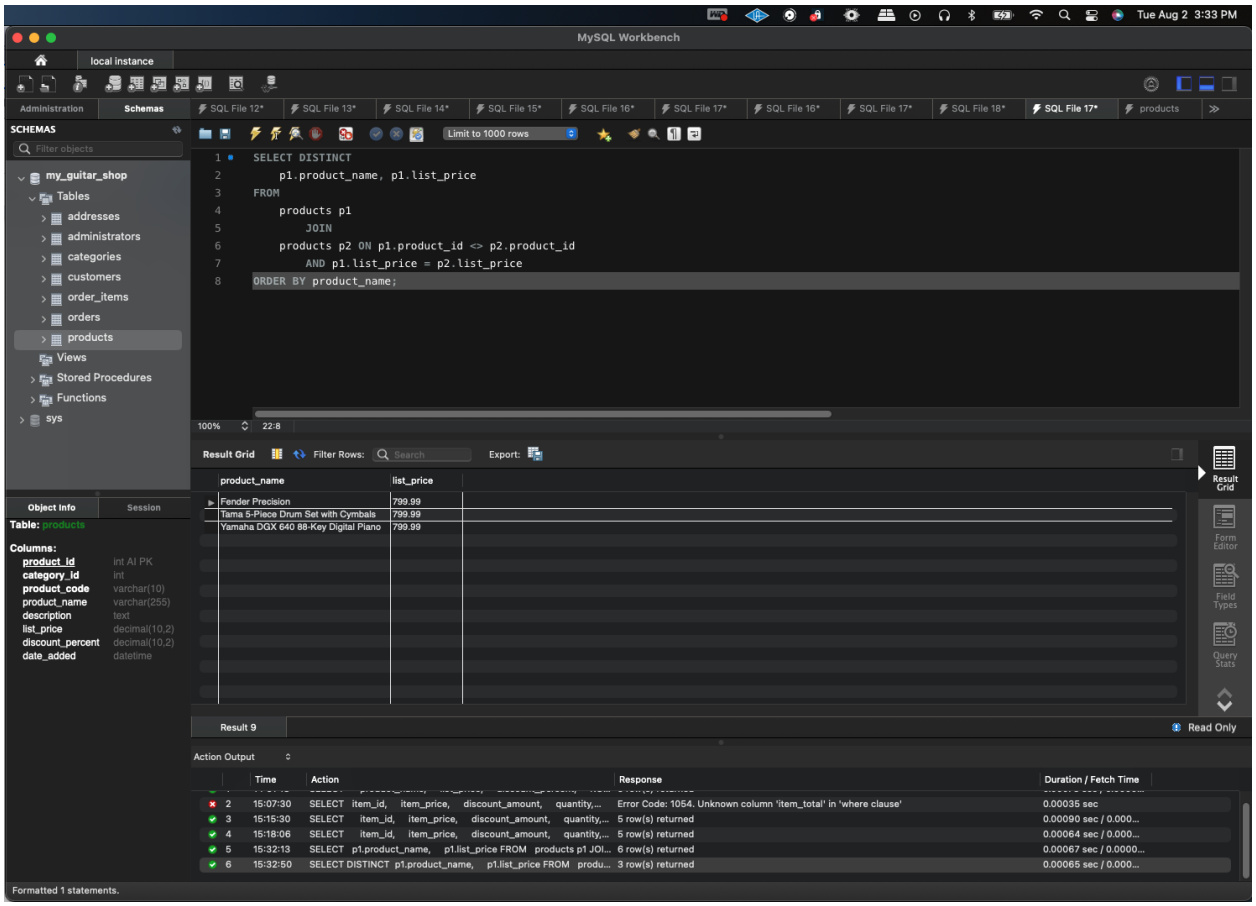

Figure 3 shows a query which uses a self join to return the distinct rows of the product name and the list price columns. The query uses aliasing and comparison operators to complete the self join and is ordered by the product name column.

Figure 4

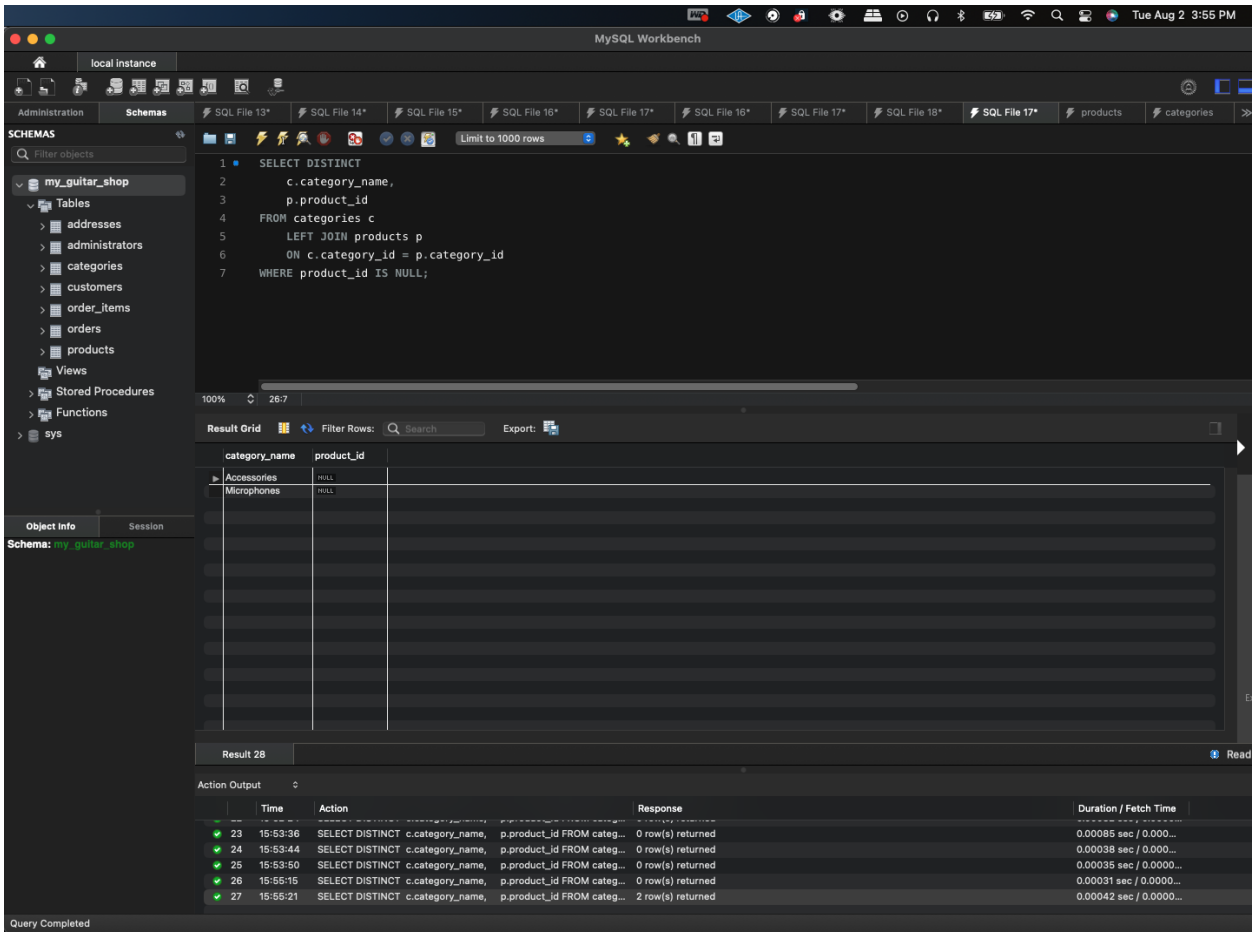

Figure 4 shows a query which selects each row of the category name and the product id column which has not been used. This is completed by an outer join on the category id column with the search condition of product id being a null value.

Figure 5

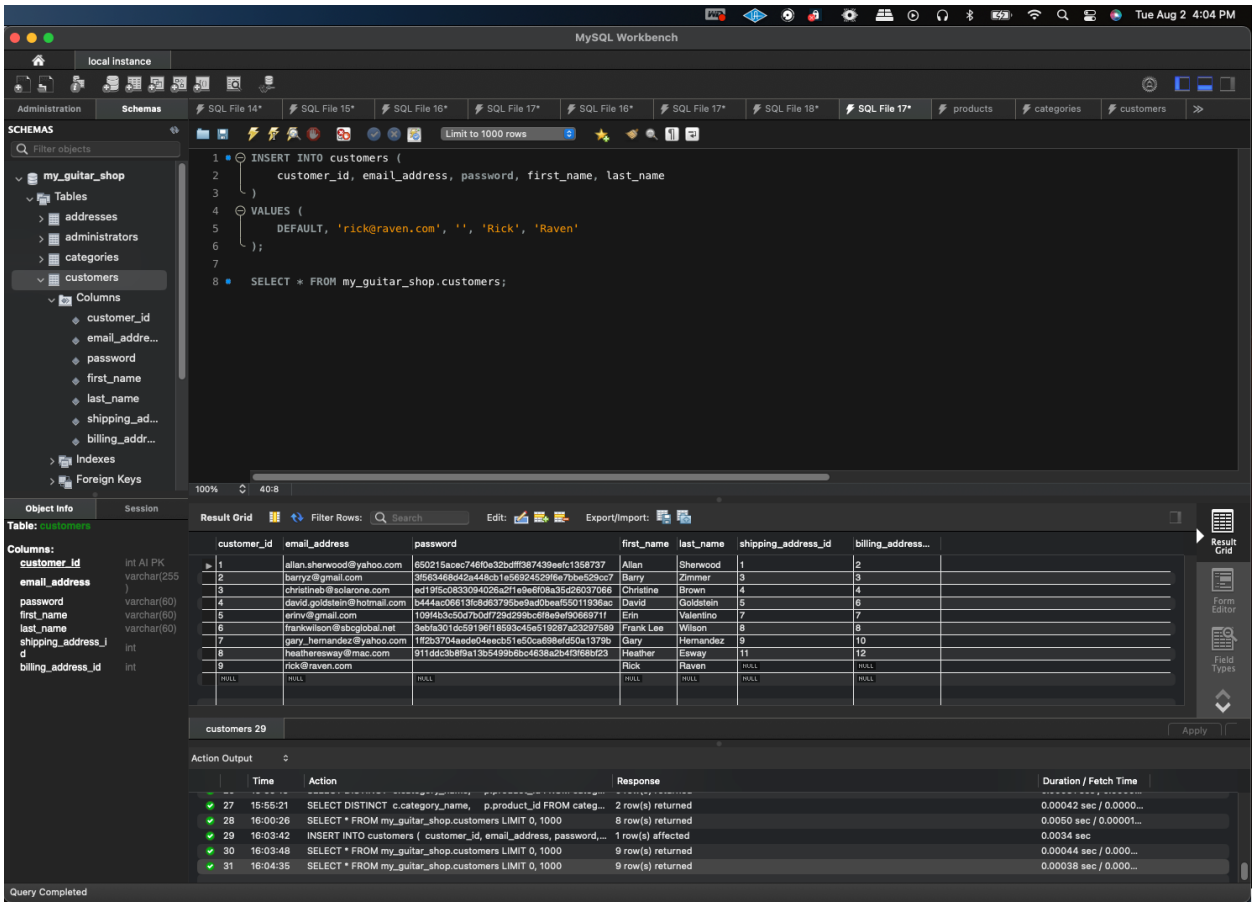

Figure 5 shows a query which inserts a row into the customers table. This insert operator defaults the customer id, has specific values for the next four columns and leaves the shipping address id and billing address id values null.

Figure 6

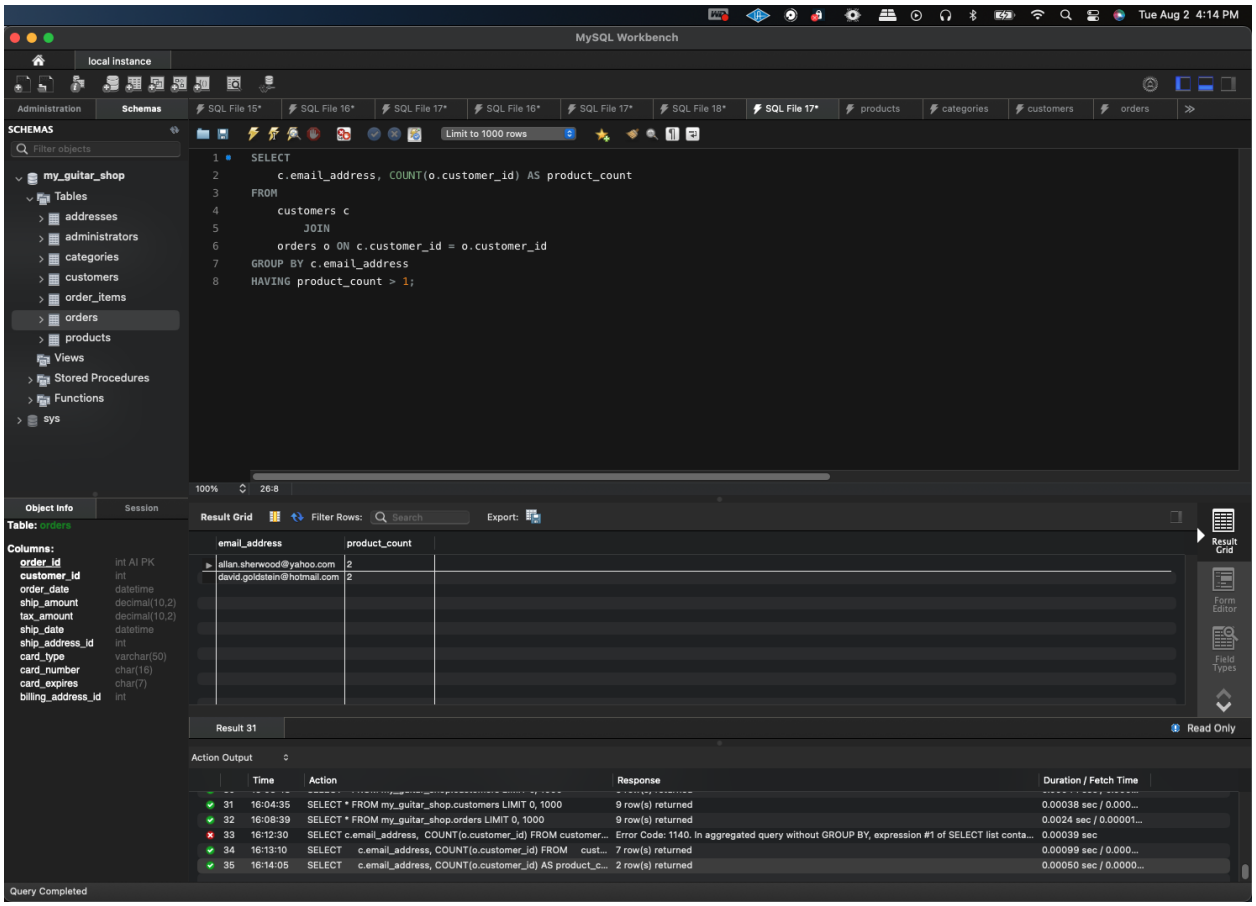

Figure 6 shows a query which selects the email address column and the count of the customer id as a column named product_count. This query joins the tables and sorts them by the email address and only displays the rows of the customers who have more than one product purchase.

Figure 7

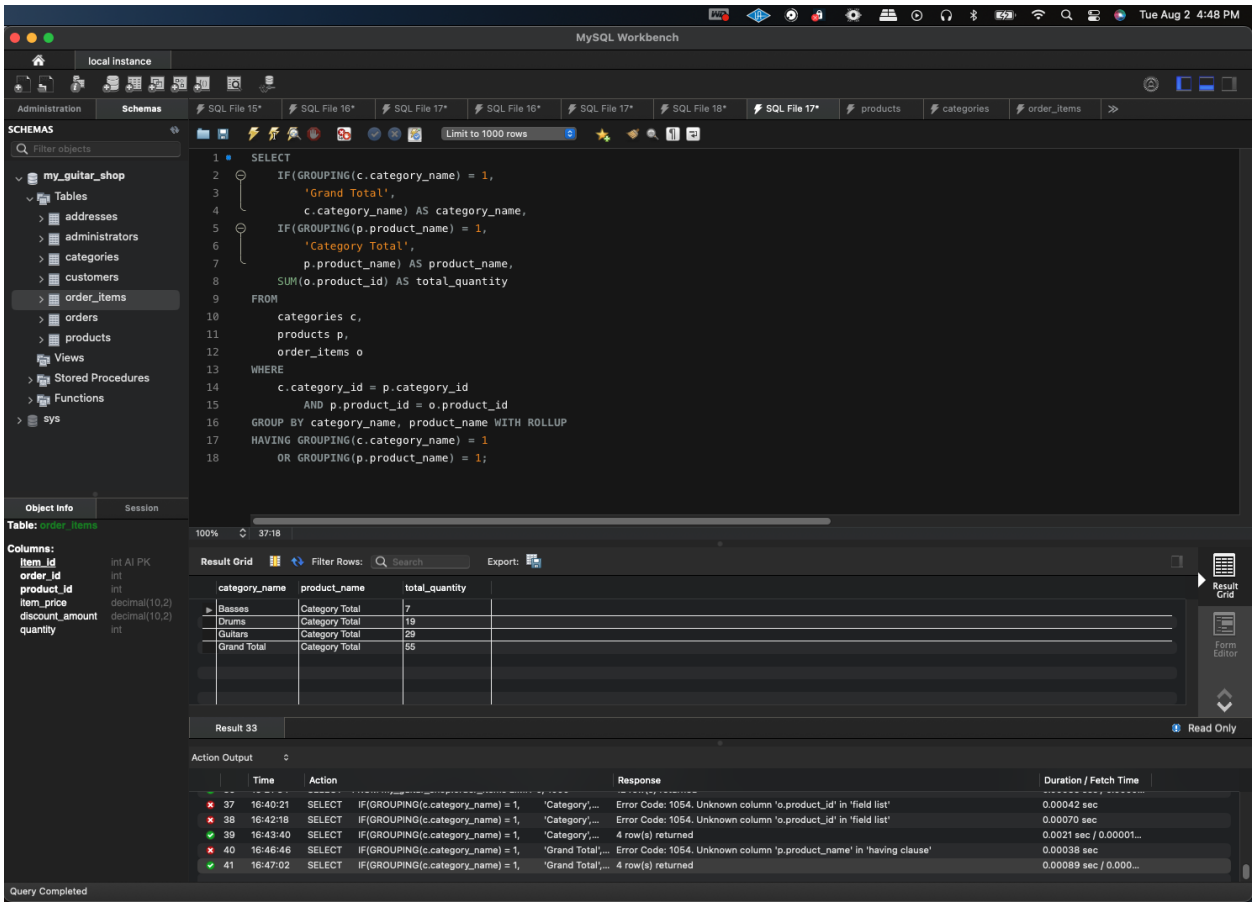

Figure 7 shows a query which selects the category name, product name and creates a column called total_quantity which is the aggregate sum of the products from each category. Essentially this query returns the total quantity of products which were purchased from each category. This also includes a grand total row which is created by the WITH ROLLUP operator to create a summary of the data and replaces the null values that would be in that row with literals.

Figure 8

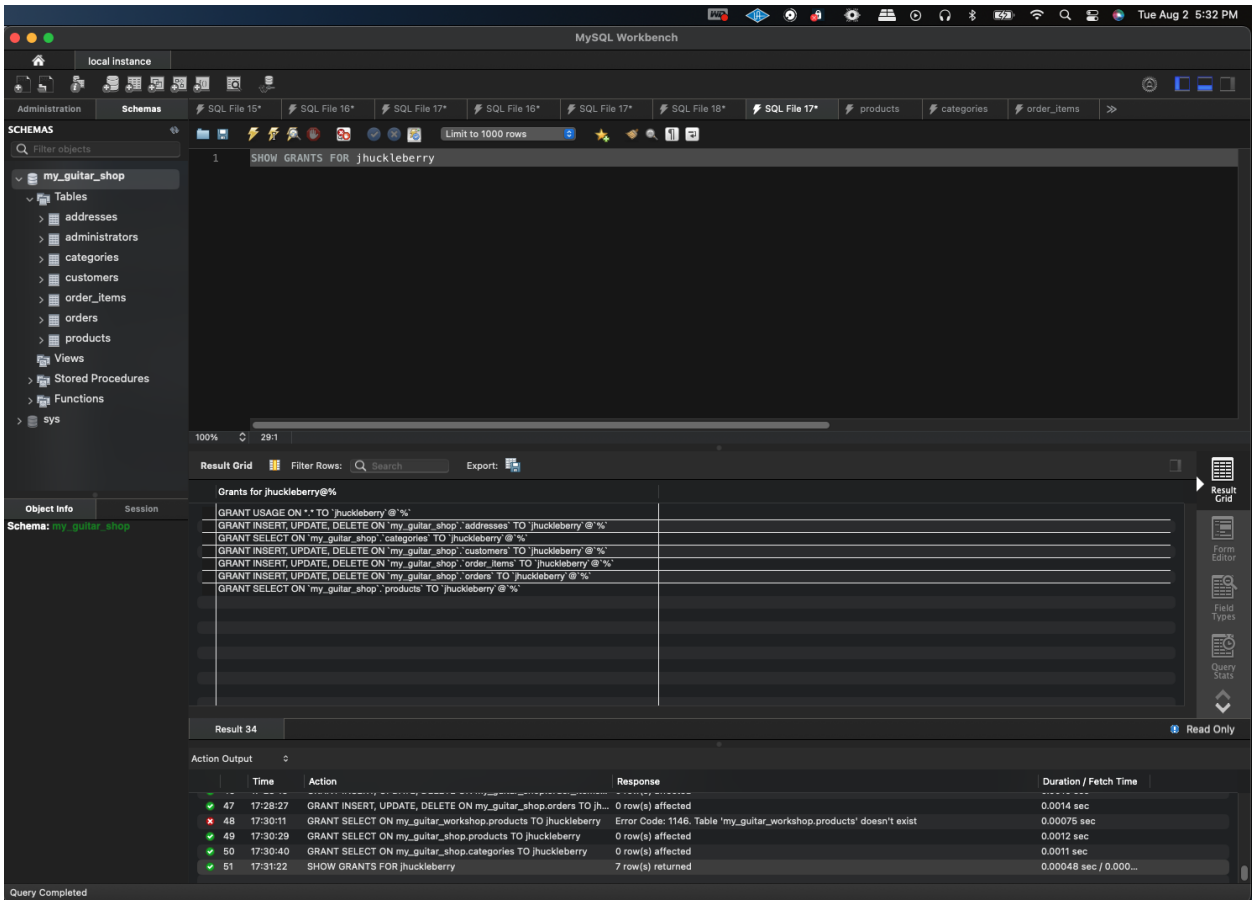

Figure 8 shows a query which shows the granted privileges to a user which was created with my first initial and last name. This user has a password and can access MySQL from any computer. The user has insert, update and delete privileges for the addresses, customers, orders and order items tables of the my guitar shop database. This user also has privileges to use the select operator on the categories and products tables within the my guitar shop database.